

B.Tech. CSE-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

TABLE OF CONTENTS

		Page Numbers
Foreword		03
VFSTR - Vision & Mission		04
CSE - AIML - Vision & Mission		04
Program Educational Objectives, Program Specific Outcomes, Program Outcomes		05
Course Structure		06
I YEAR I SEMESTER		
22MT103	Linear Algebra and Ordinary Differential Equations	13
22PY105	Semiconductor Physics and Electromagnetics	15
22EE101	Basics of Electrical and Electronics Engineering	17
22CS103	IT Workshop and Tools	19
22TP103	Programming in C	22
22EN102	English Proficiency and Communication Skills	32
22TP101	Constitution of India	34
22SA101	Physical Fitness, Sports and Games – I	
I YEAR II SEMESTER		
22MT105	Advanced Engineering Mathematics	36
22MT107	Discrete Mathematical Structures	38
22ME101	Engineering Graphics	40
22TP104	Basic Coding Competency	42
22EN104	Technical English Communication	51
22CS104	Python Programming	54
22SA102	Orientation Session	
22SA103	Physical Fitness, Sports and Games – II	
II YEAR I SEMESTER		
22ST203	Probability Theory and Statistics for Machine Learning	61
22MS201	Management Science	63
22TP202	Data Structures using Python	65
22AM201	Artificial Intelligence	70
22AM202	Digital Logic and Computer Organization	73
22CS201	Database Management Systems	76
22CT201	Environmental Studies	79
22SA201	Life Skills – I	
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication	
II YEAR II SEMESTER		
22TP203	Advanced Coding Competency	81
22TP204	Professional Communication	84
22AM203	Design and Analysis of Algorithms using Python	86
22AM204	Machine Learning	89
22CS203	Object-Oriented Programming through JAVA	92
22CS207	Operating Systems	96
22SA202	Life Skills - II	

III YEAR I SEMESTER		
22TP301	Soft Skills Laboratory	101
22AM301	Deep Learning	103
22CS204	Computer Networks	106
22DS203	Formal Languages and Automata Theory	108
22AM302	Inter-Disciplinary Project - Phase I	
22AM303	Industry Interface Course	
	Department Elective - 1	
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication	
III YEAR II SEMESTER		
22TP302	Quantitative Aptitude and Logical Reasoning	111
22AM304	Fundamentals of Image Processing	113
22AM305	Reinforcement Learning	115
22CS303	Web Technologies	117
22AM306	Inter-Disciplinary Project - Phase II	
	Department Elective - 2	
IV YEAR I SEMESTER		
22AM401	Knowledge Representation and Reasoning	123
22AM402	Text Mining	128
22CS402	Big Data Analytics	130
	Department Elective - 3	
	Department Elective - 4	
IV YEAR II SEMESTER		
22AM403	Internship / Project Work	
DEPARTMENT ELECTIVES		
22AM801	Cloud Computing	135
22AM802	Compiler Design	137
22AM803	Computer Vision	140
22AM804	Cryptography and Network Security	143
22AM805	Data Science using Python	145
22AM806	Data Warehousing and Data Mining	148
22AM807	Nature Inspired Computing Methods	151
22AM808	Soft Computing	153
22AM809	Introduction to Software Engineering	155
22AM810	Internet of Things	159
22CS801	Advanced Data Structures	161
22CS802	Advanced JAVA Programming	165
22CS803	Computer Graphics	167
22CS808	Mobile Application Development	171
HONOURS FOR CSE - AIML		
22AM951	Optimization Techniques	179
22AM952	Evolutionary Computing	182
22AM953	Kernel Methods for Pattern Analysis	184
22AM954	Parallel and Distributed Computing	186
22AM956	Capstone Project	

FOREWORD

The era of Artificial Intelligence & Machine Learning is emerging after two happening decades of Google, social media and smart mobile usage has penetrated massively into our day-to-day life; AI is going to be the next Big Thing in technological revolution. As per the 2019 Gartner report, already there is a 270% of growth was observed in the usage of AI & ML in day-to-day life from the year 2015 to 2019.

This B. Tech. CSE - AI&ML program is aimed to offer the required skills to the students on a variety of applications that can be built using AI/ ML methods, tools and techniques. Proficiency in Mathematics and Statistics will be more beneficial to the students to understand theoretical and practical design and implementation of algorithms used in the exploring fields such as Neural Networks, Natural Language Processing, Robotics, Computer Vision and Reasoning etc.

Further in this program, students will undergo the basic courses include Data Handling & Visualization, Machine Learning, Artificial Intelligence, Computer Vision, Bigdata Analytics etc. Ample AI career opportunities are present owing to wide applications in different fields. The AI&ML career opportunities include Data Analyst, Data Engineer, Machine Learning Engineer and Business Intelligence Engineer in addition to all Computing Job Profiles.

Salient Features of R 22 Curriculum:

- *Multidisciplinary holistic education with continuous learning and continuous assessment.*
- *Lateral entry and lateral exit options.*
- *Credit Earning by credit transfer.*
- *Honors/ Research Honors/ Minor/ Add-on Diploma/ Add-on Certification and Dual B.Tech.+ M.Tech./ MBA Degree of 5 Years.*
- *Semester drop option to pursue innovation, incubation, entrepreneurial and advanced exploratory activities and subsequent re-entry.*

Emphasis on continuous formative assessment with a creative summative assessment will facilitate the student to “Move away from high stake examinations – towards more continuous and comprehensive evaluation”.

The Board of Studies of B.Tech. CSE - AIML Programme consists of a right mix of eminent personalities from Academic, Research and Industry Organizations, besides experienced faculty members of the University.

External BoS Members:

1. Prof.C.R.Rao, Professor SCIS,University of Hyderabad.
2. Prof.R.V.B.Subramanyam, Professor, Department of CSE, NIT Warangal.
3. Dr.B.VenkataRamana, Assoc. Professor &HoD, Department of CSE, IIT Tirupathi.
4. Dr.V. Radha, Assoc. Professor, IDRBT, Hyderabad.
5. Dr.Nagesh Bhattu Sristy, Asst. Professor, Department of CSE, NIT AP.
6. Dr.M.Dinesh, Research Scientist, Philips, Bangalore.

I thank all the BOS members, Academic Council Members and University authorities for their continuous support and encouragement towards design of this innovative curriculum for CSE.

Dr. Venkatesulu Dondeti
Head, Department of CSE
VFSTR Deemed to be University



VIGNAN'S

Foundation for Science, Technology & Research

(Deemed to be **UNIVERSITY**)

-Estd. u/s 3 of UGC Act 1956

VISION

To evolve as a centre of high repute in Computer Science & Engineering and create computer software professionals trained on problem solving skills imbued with ethics to serve the ever evolving and emerging requirements of IT Industry and society at large.

MISSION

- Imparting quality education through well designed curriculum, innovative teaching and learning methodologies integrated with professional skill development activities to meet the challenges in the career.
- Nurture research and consultancy activities amongst students and faculty by providing State-of-art facilities and Industry-Institute Interaction.
- Developing capacity to learn new technologies and apply to solve social and industrial problems to become an entrepreneur.

Department of CSE-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

VISION of the department

To evolve as a centre of high repute in Computer Science & Engineering and create computer software professionals trained on problem solving skills imbued with ethics to serve the ever evolving and emerging requirements of IT Industry and society at large.

MISSION of the department

- M₁:** Imparting quality education through well designed curriculum, innovative teaching and learning methodologies integrated with professional skill development activities to meet the challenges in the career.
- M₂:** Nurture research and consultancy activities amongst students and faculty by providing State-of-art facilities and Industry-Institute Interaction.
- M₃:** Developing capacity to learn new technologies and apply to solve social and industrial problems to become an entrepreneur.

B.Tech in CSE - Artificial Intelligence and Machine Learning

Program Educational Objectives (PEOs)

Program Educational Objectives (PEOs) are established through a consultation process. PEOs are broad statements that describe the career and professional accomplishments that the graduates should achieve within three to five years after their graduation.

Graduates of the UG-CSE-AI&ML program will be able to

- PEO1:** Pursue a successful professional career in IT and IT-enabled industries.
- PEO2:** Pursue lifelong learning in generating innovative engineering solutions using research and complex problem-solving skills.
- PEO3:** Demonstrate professionalism, ethics, inter-personal skills and continuous learning to develop leadership qualities.

Program Specific Outcomes (PSOs)

The students will be able to

- PSO1: Data Science for Life:** Ability to represent the knowledge and predicate logic and then transform the real-life information into a different representation.
- PSO2: AI Application Development:** Design and Develop Deep Learning Models on the Cloud System using Cloud Services like Amazon Web Services, Microsoft Azure, Hadoop System, etc., to work on AI & ML for providing solutions to Geo-Socio-Economic problems.

Program Outcomes (POs)

Program Outcomes (POs), are attributes acquired by the student at the time of graduation. The POs given in below, ensure that the POs are aligned to the Graduate Attributes (GAs) specified by National Board of Accreditation (NBA).

- PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- PO2: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- PO9: Individual and team-work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

R22 B.Tech.

4
YEARDEGREE
PROGRAMME

COURSE STRUCTURE - R22

I Year I Semester

Course Code	Course Title	L	T	P	C	Course category
22MT103	Linear Algebra and Ordinary Differential Equations	3	2	0	4	Basic Sciences
22PY105	Semiconductor Physics and Electromagnetics	2	0	2	3	Basic Sciences
22EE101	Basics of Electrical and Electronics Engineering	2	0	2	3	Basic Engineering
22CS103	IT Workshop and Tools	0	2	4	3	Basic Engineering
22TP103	Programming in C	2	0	4	4	Basic Engineering
22EN102	English Proficiency and Communication Skills	0	0	2	1	Humanities
22TP101	Constitution of India	0	2	0	1	Binary Grade
22SA101	Physical Fitness, Sports and Games-I	0	0	3	1	Binary Grade
Total		9	6	17	20	
		32 Hrs			20	

I Year II Semester

Course Code	Course Title	L	T	P	C	Course category
22MT105	Advanced Engineering Mathematics	3	2	0	4	Basic Sciences
22MT107	Discrete Mathematical Structures	2	2	0	3	Basic Sciences
22ME101	Engineering Graphics	2	0	2	3	Basic Engineering
22TP104	Basic Coding Competency	0	1	3	2	Basic Engineering
22EN104	Technical English Communication	2	0	2	3	Humanities
22CS104	Python Programming	2	0	2	3	Professional Core
22SA102	Orientation Session	0	0	6	3	Binary Grade
22SA103	Physical Fitness, Sports and Games – II	0	0	3	1	Binary Grade
Total		11	5	18	22	
		34 Hrs			22	

COURSE STRUCTURE - R22

R22 B.Tech.

4

YEAR
**DEGREE
PROGRAMME**


II Year I Semester

Course Code	Course Title	L	T	P	C	Course category
22ST203	Probability Theory and Statistics for Machine Learning	3	0	2	4	Basic Sciences
22MS201	Management Science	2	2	0	3	Basic Engineering
22TP202	Data Structures using Python	2	2	2	4	Humanities
22AM201	Artificial Intelligence	2	2	2	4	Professional Core
22AM202	Digital Logic and Computer Organization	2	2	0	3	Professional Core
22CS201	Database Management Systems	2	2	2	4	Professional Core
22CT201	Environmental Studies	1	1	0	1	Basic Sciences
22SA201	Life Skills-I	0	0	2	1	Binary Grade
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication.	0	0	0	1	Binary Grade
Total		14	11	10	25	
		35 Hrs			25	

II Year II Semester

Course Code	Course Title	L	T	P	C	Course category
22TP203	Advanced Coding Competency	0	0	2	1	Basic Engineering
22TP204	Professional Communication	0	0	2	1	Humanities
22AM203	Design and Analysis of Algorithms using Python	2	2	2	4	Professional Core
22AM204	Machine Learning	3	0	2	4	Professional Core
22CS203	Object-Oriented Programming through JAVA	2	0	4	4	Professional Core
22CS207	Operating Systems	2	0	2	3	Professional Core
22SA202	Life Skills - II	0	0	2	1	Binary Grade
	Open Elective – 1	3	0	0	3	Open Elective
Total		12	2	16	21	
	Minor / Honours - 1	3	0	2	4	
Total		15	2	18	25	
		35 Hrs			25	

R22 B.Tech.

4
YEARDEGREE
PROGRAMME

COURSE STRUCTURE - R22

III Year I Semester

Course Code	Course Title	L	T	P	C	Course category
22TP301	Soft Skills Laboratory	0	0	2	1	Humanities
22AM301	Deep Learning	3	0	2	4	Professional Core
22CS204	Computer Networks	3	0	2	4	Professional Core
22DS203	Formal Languages and Automata Theory	2	2	0	3	Professional Core
22AM302	Inter-Disciplinary Project – Phase I	0	0	2	0	Project
22AM303	Industry Interface Course	1	0	0	1	Binary Grade
	Department Elective – 1	3	0	2	4	Department Elective
	Open Elective – 2	3	0	0	3	Open Elective
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication.	0	0	0	1	Binary Grade
Total		15	2	10	21	
	Minor / Honours - 2	3	0	2	4	
Total		18	2	12	25	
				32 Hrs	25	

III Year II Semester

Course Code	Course Title	L	T	P	C	Course category
22TP302	Quantitative Aptitude and Logical Reasoning	1	2	0	2	Humanities
22AM304	Fundamentals of Image Processing	2	0	2	3	Professional Core
22AM305	Reinforcement Learning	2	2	0	3	Professional Core
22CS303	Web Technologies	2	0	4	4	Professional Core
22AM306	Inter-Disciplinary Project – Phase II	0	0	2	2	Project
	Department Elective – 2	3	0	2	4	Department Elective
	Open Elective – 3	3	0	0	3	Open Elective
Total		13	4	10	21	
	Minor / Honours - 3	3	0	2	4	
Total		16	4	12	25	
				32 Hrs	25	

COURSE STRUCTURE - R22

R22 B.Tech.

4

YEAR
**DEGREE
PROGRAMME**


IV Year I Semester

Course Code	Course Title	L	T	P	C	Course category
22AM401	Knowledge Representation and Reasoning	2	2	0	3	Professional Core
22AM402	Text Mining	3	0	2	4	Professional Core
22CS402	Big Data Analytics	3	0	2	4	Professional Core
	Department Elective – 3	3	0	2	4	Department Elective
	Department Elective – 4	3	0	2	4	Department Elective
	Total	14	2	8	19	
	Minor / Honours – 4	3	0	2	4	
	Total	17	2	10	23	
		29 Hrs			23	

IV Year II Semester

Course Code	Course Title	L	T	P	C	Course category
22AM403	Internship / Project Work	0	2	22	12	Project
	Total	0	2	22	12	
	Minor / Honors – 5	0	2	6	4	
	Total	0	4	28	16	
		32 Hrs			16	

R22 B.Tech.

4
YEARDEGREE
PROGRAMME

COURSE STRUCTURE - R22

Department Electives

Course Code	Course Title	L	T	P	C
22AM801	Cloud Computing	3	0	2	4
22AM802	Compiler Design	3	2	0	4
22AM803	Computer Vision	3	0	2	4
22AM804	Cryptography and Network Security	3	0	2	4
22AM805	Data Science using Python	2	2	2	4
22AM806	Data Warehousing and Data Mining	3	0	2	4
22AM807	Nature Inspired Computing Methods	3	2	0	4
22AM808	Soft Computing	3	2	0	4
22AM809	Introduction to Software Engineering	3	0	2	4
22AM810	Internet of Things	3	0	2	4
22CS801	Advanced Data Structures	2	2	2	4
22CS802	Advanced JAVA Programming	2	2	2	4
22CS803	Computer Graphics	2	2	2	4
22CS808	Mobile Application Development	2	0	4	4

Honours for CSE

Course Code	Course Title	L	T	P	C
22AM951	Optimization Techniques	3	2	0	4
22AM952	Evolutionary Computing	3	2	0	4
22AM953	Kernel Methods for Pattern Analysis	3	0	2	4
22AM954	Parallel and Distributed Computing	3	2	0	4
22AM956	Capstone Project	0	2	6	4

I
YEAR

B.Tech.

COURSE CONTENTS

I SEM & II SEM

CSE - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

I SEMESTER

- ▶ 22MT103 - Linear Algebra and Ordinary Differential Equations
- ▶ 22PY105 - Semiconductor Physics and Electromagnetics
- ▶ 22EE101 - Basics OF Electrical and Electronics Engineering
- ▶ 22CS103 - IT Workshop and Tools
- ▶ 22TP103 - Programming in C
- ▶ 22EN102 - English Proficiency and Communication Skills
- ▶ 22TP101 - Constitution of India
- ▶ 22SA101 - Physical Fitness, Sports and Games-I

II SEMESTER

- ▶ 22MT105 - Advanced Engineering Mathematics
- ▶ 22MT107 - Discrete Mathematical Structures
- ▶ 22ME101 - Engineering Graphics
- ▶ 22TP104 - Basic Coding Competency
- ▶ 22EN104 - Technical English Communication
- ▶ 22CS104 - Python Programming
- ▶ 22SA102 - Orientation Session
- ▶ 22SA103 - Physical Fitness, Sports and Games-II

22MT103 LINEAR ALGEBRA AND ORDINARY DIFFERENTIAL EQUATIONS

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Basics of matrices, Differentiation and Integration.

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to build a grasp of the principles of mathematics through matrices, differential equations and applications that serves as an essential tool in several engineering applications.

MODULE-1

UNIT-1

12L+8T+0P=20 Hours

MATRICES:

Definition of matrix; Types of matrices; Algebra of matrices, adjoint of a matrix, inverse of a matrix through adjoint and elementary row operations, Rank of a matrix, Echelon form, Normal form. Eigen values and Eigen vectors (up to 3 x 3 matrices only) and properties (without proofs).

UNIT-2

12L+8T+0P=20 Hours

APPLICATIONS OF MATRICES:

Consistency of system of linear equations, Solution of system of linear equations having unique solution and involving not more than three variables by Gauss elimination method and Gauss Jordan method. Cayley-Hamilton theorem (without proof), Power of a matrix, Inverse of a matrix. Strength of materials and strength of beams using Eigen value and Eigen vectors.

PRACTICES:

- Compute inverse of a matrix if exists.
- Explain with suitable examples how rank of matrix is independent of the elementary operations.
- Explain with suitable examples how rank of matrix is unique.
- Discuss with suitable examples when eigen values and eigen vectors are possible for a matrix.
- Discuss the possibility of solution of a system of equations.
- Discuss when inverse and power of a matrix exist using Cayley - Hamilton theorem.

MODULE-2

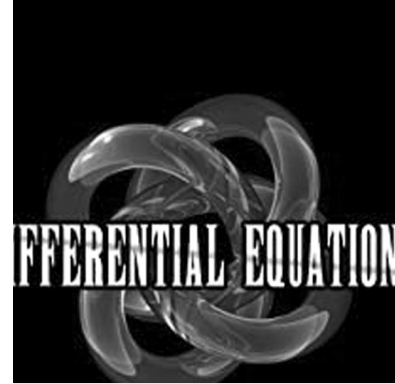
UNIT-1

12L+8T+0P=20 Hours

ORDINARY DIFFERENTIAL EQUATIONS (ODE):

First Order Differential Equations: Introduction to ODE, variable separable method, homogenous and non-homogenous differential equations, linear differential equations, Bernoulli's equations.

Second Order Differential Equations: Linear differential equations with constant coefficients with RHS of the form e^{ax} , x^n , $\sin(ax)$ or $\cos(ax)$.



Source: <https://www.amazon.com/Differential-Equations/dp/B01H30X2JA>

SKILLS:

- ✓ Find rank of a matrix using different methods.
- ✓ Compute the eigen values and eigen vectors of a matrix.
- ✓ Find analytical solution of a differential equation using appropriate method.
- ✓ Demonstrate any one numerical method to solve differential equation

UNIT-2**12L+8T+0P=20 Hours****APPLICATIONS OF ODE:****Applications of ODE:** Newton's law of cooling, Law of natural growth and decay, LR Circuit.**PRACTICES:**

- Check the order and degree of an ODE.
- Find solution for any four ordinary differential equations by applying suitable method.
- Find numerical solution for any four ordinary differential equations by applying suitable method.
- Discuss some applications of ODE.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the concepts of rank, eigen values and eigenvectors of a matrix and finding inverse of a matrix and powers of a matrix.	Apply	1	1, 2, 9, 10, 12
2	Apply differential equations in real life problems.	Apply	2	1, 2, 9, 10, 12
3	Analyse the solution of a system of linear equations and find it.	Analyze	1	1, 2, 9, 10, 12
4	Inspect the analytical method for solving differential equations and applications.	Analyze	2	1, 2, 9, 10, 12

TEXT BOOKS:

1. N. P. Bali, K. L. Sai Prasad, "A Textbook of Engineering Mathematics I, II, III", 2nd edition Universal Science Press, New Delhi, 2018.
2. B. S. Grewal, "Higher Engineering Mathematics", 44th edition, Khanna Publishers, 2018.

REFERENCE BOOKS:

1. Erwin Kreyszig, "Advanced Engineering Mathematics", 10th Edition, John Wiley & Sons, Inc, 2015.
2. H. K. Dass and Er. Rajanish Verma, "Higher Engineering Mathematics", 3rd revised edition, S. Chand & Co., 2015.
3. B. V. Ramana, "Advanced Engineering Mathematics", TMH Publishers, 2020.
4. T. K.V. Iyengar et al, "Engineering Mathematics, I, II, III", S. Chand & Co., New Delhi.

22PY105 SEMICONDUCTOR PHYSICS AND ELECTROMAGNETICS

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Basics of vectors and semiconductors.

COURSE DESCRIPTION AND OBJECTIVES:

This course ensures commensurable understanding of electrostatics and magnetostatics. It enunciates the electron dynamics in solids through the conceptual grasp of principles of quantum mechanics. This embark perspective outlook on optoelectronic devices and optical fibres in the backdrop of semiconductor physics.

MODULE-1

UNIT-1

10L+0T+10P=20 Hours

ELECTROSTATIC AND MAGNETOSTATICS:

Electrostatics: Introduction to Vector analysis, Computation of electric field and potential due to Point charge, linear charge density, surface charge density, bulk charge density, Coulomb's law, Electric field due to line of charges, Gauss law, Differential Form of Gauss law, Applications, Electric field due to a charged sphere – inside, on the surface, and outside, Electric field due to a spherical shell- inside and outside.

Magnetostatics: Introduction to magnetic force – Lorentz force, Biot-Savart's law, Magnetic field due to a linear conductor – magnetic field due to a circular loop – Ampere's law, Faraday's law in integral form; Lenz's law, Maxwell's equations – correction to Ampere's law.

UNIT-2

6L+0T+6P=12 Hours

QUANTUM MECHANICS AND FREE ELECTRON THEORY:

Quantum mechanics: Introduction to Quantum mechanics; Concepts of wave and particle duality of radiation; de Broglie's concepts of matter waves, Schrödinger's time-independent wave equation – Eigen values and Eigen functions; Particle confined in a one-dimensional infinite Potential square well.

Free electron theory of solids: Classical and Quantum free electron theory of metals; Fermi-Dirac distribution; Density of states – derivation -Bloch's Theorem (Qualitative); Classification of solids based on energy bands.

PRACTICES:

- Photoelectric effect-Determination of Planck's constant.
- Stewart & Gee's Experiment- Study of magnetic field along the axis of a current carrying coil.
- Melde's Experiment - determination of the frequency of tuning fork.
- Sonometer- Determination of AC frequency.

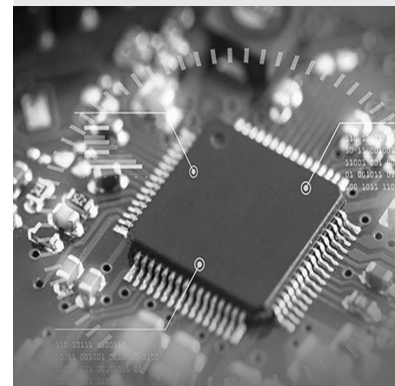
MODULE-2

UNIT-1

8L+0T+8P=16 Hours

SEMICONDUCTOR PHYSICS AND OPTOELECTRONICS:

Introduction, Classification of Semiconductors, Direct and indirect band gap semiconductors, Intrinsic semiconductors; Variation of Intrinsic carrier concentration with temperature, Fermi level, and conductivity; Extrinsic semiconductor, the effect of temperature on carrier concentration in extrinsic semiconductors, Band diagrams of extrinsic semiconductors; Hall effect, Classification of optoelectronic devices; Photo voltaic cell, LED.



Source: <https://www.scitusacademics.com/product/semiconductor-devices-and-circuits/>

SKILLS:

- ✓ Able to compute the electric and magnetic field and potentials in different applications
- ✓ Apply the quantum laws to understand the electron dynamics of solids
- ✓ Realizing the importance of optoelectronic devices

UNIT-2**8L+0T+8P=16 Hours****LASERS AND OPTICAL FIBERS:**

Introduction to lasers, Population inversion & pumping processes, Semiconductor diode laser, Applications of lasers. Optical fiber-Numerical Aperture, types of optical fibres, Fiber optic communication system.

PRACTICES:

- Laser - Determination of wavelength.
- Optical fiber - Determination of Numerical aperture – Acceptance angle.
- Determination of Energy Band gap of p-n junction diode.
- Hall Effect - Determination of Hall coefficient.
- Solar cell – Determination of Fill factor & efficiency.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply Maxwell's equations to unravel electron dynamics in amidst of electric and magnetic fields.	Apply	1	1, 2, 4, 5, 9, 10
2	Discriminate solids based on principles of quantum mechanics.	Analyse	1	1, 2, 3, 4, 9, 10
3	Assessment of semiconductors in the perspective of optoelectronic devices.	Evaluate	2	1, 3, 4, 5, 6, 9, 10
4	Comprehend the knowledge of Lasers and optical fibers to conceive their applications in vivid domains.	Apply	2	1, 2, 3, 5, 9, 10

TEXT BOOKS:

1. S.O. Pillai, "Solid State Physics", New age International publishers, 8th edition, 2018.
2. H.C. Varma, "Classical Electromagnetism", Bharathi Bhavan Publication, 2022.

REFERENCE BOOKS:

1. D. Halliday, R. Resnick and J. Walker, "Fundamentals of Physics", 6th edition, John Wiley and Sons, New York, 2001.
2. M.N. Avadhanulu, "Engineering Physics", S. Chand publications 2010.
3. Charles Kittel, "Introduction to Solid State Physics", 7th edition, Wiley, Delhi, 2007.
4. Donald A. Neamen, "Semiconductor Physics and Devices: Basic Principle", 4th edition, McGraw-Hill, New York, 2012.
5. David J. Griffiths, "Introduction to Electrodynamics", 3rd edition, Prentice Hall of India, New Delhi, 2012.
6. N.W. Ashcroft and N.D. Mermin, "Solid State Physics", International student edition, Brooks Cole, 2008.

22EE101 BASICS OF ELECTRICAL AND ELECTRONICS ENGINEERING

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Electrostatics and Electromagnetism.

COURSE DESCRIPTION AND OBJECTIVES:

This course provides an insight into the functioning of basic electrical components like resistor, inductor and capacitor. It deals with the constructional and operational details of AC machines. It also deals with the basic electronic components like P-N junction diode, Zener diode, Transistor and their characteristics.

MODULE-1

UNIT-1

8L+0T+8P=16 Hours

FUNDAMENTALS OF ELECTRIC CIRCUITS:

DC Circuits: Concept of network, Active and passive elements, Voltage and current sources, Concept of linearity and linear network, Unilateral and bilateral elements, R, L and C as linear elements, Ohm's Law, Kirchhoff's Laws, Application to simple series, Parallel circuits, Mesh and nodal analysis of resistive circuits with DC source.

AC circuits: Generation of AC voltage, Frequency, Average value, R.M.S. value, Form factor, Peak factor for sinusoidal only.

UNIT-2

8L+0T+8P=16 Hours

SEMICONDUCTOR DEVICES:

Classification of semiconductors, P-N junction diode -operation and its characteristics, Half wave rectifier - operation, efficiency; Full wave rectifiers -types, operation, efficiency; Zener diode and its characteristics, Zener diode as Voltage regulator.

Bi polar junction transistor- operation, types (NPN & PNP)

PRACTICES:

- Verification of Ohm's law.
- Verification of Kirchhoff's current law.
- Verification of Kirchhoff's voltage law.
- Determination of R.M.S. Values of sinusoidal waveform.
- Verification of PN junction diode characteristics under both forward and reverse bias.
- Verification of Zener diode characteristics under reverse bias.

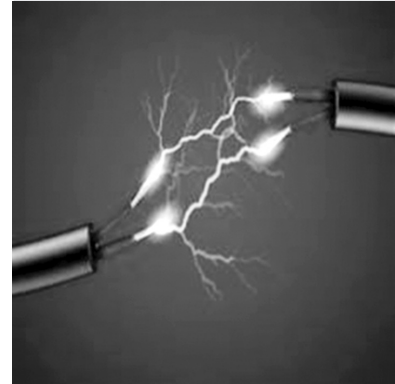
MODULE-2

UNIT-1

8L+0T+8P=16 Hours

ANALYSIS OF AC CIRCUITS:

Analysis of single- phase ac circuits consisting of R, L, C, RL, RC (series and parallel) (simple numerical problems). Introduction to three phase system, Relation between phase and line quantities of voltages and currents in star and delta connected systems (Elementary treatment only).



Source: <https://vita.vision.org.in/emerging-technologies-in-electrical-engineering/>

SKILLS:

- ✓ Distinguish between linear and nonlinear elements by looking at VI characteristics.
- ✓ Develop a simple loop generator.
- ✓ Design a voltage regulator using Zener diode.
- ✓ Design a half wave rectifier using PN junction diode.
- ✓ Design a full wave rectifier using PN junction diodes.

UNIT-2**8L+0T+8P=16 Hours****AC MACHINES:**

Electromagnetism: Concepts of Magneto motive force, Reluctance, Flux and flux density, Concept of self-inductance and mutual inductance, Coefficient of coupling .

Static AC Machine: Principle of operation of single phase transformer, Constructional features, EMF equation (simple numerical problems).

Rotating AC Machine Principle of operation of three phase induction motor, Slip ring and squirrel cage motors, Torque equation; Constructional details of synchronous machine.

PRACTICES:

- Transformation ratio of a single phase transformer at different loads.
- Measurement of Energy in single phase resistive load circuit.
- Measurement of Power in single phase resistive load circuit.
- Determination of impedance in complex AC circuits.
- Verification of line and phase quantities in a balanced three phase system.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the resistive circuits with independent sources and find its solution.	Analyze	1, 2	1, 2, 6, 9
2	Solve the AC (single and three phase) and DC circuits using different methods.	Apply	1, 2	1, 2, 9, 12
3	Apply the concepts of electromagnetism for its applications.	Apply	2	1, 2, 3, 9, 12
4	Examine the different electrical equipment.	Evaluate	2	1, 2, 9, 12
5	Acquire the knowledge of semiconductor devices to create circuits.	Create	1	1, 2, 3, 9, 12

TEXT BOOKS:

1. V. K. Mehta, "Principles of Electrical Engineering and Electronics", 1st edition, S.Chand & Co., Publications, New Delhi, 2019.
2. D.P. Kothari, "Basic Electrical and Electronics Engineering", 2nd edition, TMH, New Delhi, 2017.

REFERENCE BOOKS:

1. Millman and Halkias, "Electronic Devices and Circuits", Mc Graw Hill, 2006.
2. A.K. Thereja and B.L. Thereja, "Electrical Technology", Vol.-II, S. Chand & Co., Publications, 2020.
3. U. Bakshi and A. Bakshi, "Basic Electrical Engineering", 1st edition, Technical Publications, Pune, Nov 2020.

22CS103 IT WORKSHOP AND TOOLS

Hours Per Week :

L	T	P	C
0	2	4	3



Source: <https://www.facebook.com/TheITWorkshopWA>

PREREQUISITE KNOWLEDGE: Basics of Computer knowledge, Applications of Computers.

COURSE DESCRIPTION AND OBJECTIVES:

This course enables the students to learn various components of a computer system, assembly and dis-assembly of various components, troubleshooting, installation of OS and other applications. Also practicing of the usage of software tools such as word, excel, ppt and LaTeX, text and image editors.

MODULE-1

UNIT-1

0L+16T+32P=48 Hours

PRACTICING EXERCISES ON HARDWARE DEVICES:

- Demo of various physical components of a computer system.
- Integration of various components of a computer system and dismantling.
- Installation of OS in a computer system through various storage devices.
- Installation of OS in a computer system through cloning.
- Demonstration of booting process of a computer system
- Detection of faulty components such as hard disk, RAM, SMPS, network interface in a computer system.
- Demonstration of program execution environment.
- Demo of Windows/Linux file system.
- Demo of location OS files in the file system (Windows/Linux).
- Configuration of network interface in a computer system and troubleshooting of network connectivity issues.
- Demo of shell scripts for maintenance and administration of a computer system.
- Usage of editor tools.
- Installation of software tools such as C compiler / interpreter, Java IDE, Python IDLE, Pycharm etc.
- Installation of antivirus software, web browsers and application of servers such as Apache server etc.

ACTIVITIES:

- Assemble and dis-assemble of various components of a computer system.
- Connect devices to various interfaces-(a) Serial Port, (b) Parallel Port, (c) USB Port, (d) Fire wire, (e) RJ45 connector, (f) VGA connector, (g) Audio plugs (Line-In, Line-Out and microphone), (h) PS/2 Port, and (h) SCSI Port.
- Install Linux/windows OS in your computer.
- Identification of different Computer hardware problems and troubleshooting of the same.
- Editing of image, audio and video files using different editor tools.
- Build Ethernet and Wi-Fi LANs.

SKILLS:

- ✓ *Integration of various components of a computer system.*
- ✓ *Trouble shooting of components of a computer system.*
- ✓ *Installation of OS and its various tools/ applications.*
- ✓ *Usage of IT tools such as MS-Word, LaTeX etc.*
- ✓ *Creating the documents using MS-Word and LaTeX.*
- ✓ *Analysing and visualizing data with excel.*
- ✓ *Developing various power point presentations*

MODULE-2**UNIT-1****0L+16T+32P=48 Hours****PRACTICING EXERCISES USING SOFTWARE TOOLS:**

- Prepare your resume using MS-word.
- Design a "Birthday Invitation" card.
- Design a Timetable given to you at the beginning of the semester without grid lines.
- Using Draw Table feature, insert a 7-column, 6-row table to create a calendar for the current month.
 - a. Enter the names of the days of the week in the first row of the table.
 - b. Centre the day names horizontally and vertically.
 - c. Change the font and font size as desired.
 - d. Insert a row at the top of the table.
 - e. Merge the cells in the row and enter the current month and year using a large font size.
 - f. Shade the row.
 - g. Enter and right-align the dates for the month in the appropriate cells of the table.
 - h. Change the outside border to a more decorative border. Identify two important dates in the calendar and shade them.
- Prepare mark sheet using MS-Excel.
- Create a pivot table to analyse your worksheet data.
- Prepare a presentation on your university using MS-PowerPoint.
- Design a Magazine cover. Use the following:
 - 1) Select a theme for the page,
 - 2) Insert either a picture or clipart, and
 - 3) Use WordArt.
- Design a poster inviting all students of your university to the Computer Festival.
- Installation and demonstration of LaTeX.
- Prepare professional pdf documents using LaTeX.
- Prepare LaTeX document containing mathematical equations.

ACTIVITIES:

- Create a 5-page document. Use hyperlinks, insert bookmarks in the same document.
- Design a worksheet using the following functions-MODE, STDDEV, VARIANCE, MEDIAN, SIN, COS, TAN, COUNT, MAX, MIN, ABS, MOD, SUM, SUMIF, POWER.
- Create bar graphs, pie charts and line charts in excel.
- Create a 5-slide presentation on any topic. Use Images, Graphs, Chart, Tables, Animation, Time, Bullets, Transition, Sound, Hyperlink, Background template, Header and Footer.
- Create a following numbered list using LaTeX.

INTRODUCTION:

This is a display of numbered list

- Abstract
- Introduction.
- Section 1.
 - a) Section 1.1.
 - b) Section 1.2.
- Section 2.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Ability to assemble and disassemble the computer system components and trouble shooting.	Apply	1	1
2	Installing Operating Systems and understanding the system booting process.	Understand	1	1
3	Ability to develop system maintenance using shell scripts.	Apply	1	1
4	Create word documents, presentations and spread sheets by applying various tools.	Create	2	2, 5

TEXT BOOKS:

1. Fundamentals of Computers by Reema Thareja, Oxford University Press 2nd edition 2019, India.
2. Stefan Kottwitz, "LaTeX Beginner's Guide: Create visually appealing texts, articles, and books for business and science using LaTeX", 2nd Edition, Kindle, 2021.

REFERENCE BOOKS:

1. Priti Sinha and Pradeep K. Sinha, "Computer Fundamentals: Concepts, Systems and Applications", 8th edition, BPB Publications, 2004.
2. John Walkenbach, Herb Tyson, Michael R.Groh and FaitheWempen, "Microsoft Office 2010 Bible", Wiley.

22TP103 PROGRAMMING IN C

Source: Techgig.com

Hours Per Week :

L	T	P	C
2	0	4	4

PREREQUISITE KNOWLEDGE: Fundamentals of Problem Solving.**COURSE DESCRIPTION AND OBJECTIVES:**

This course is aimed to impart knowledge on basic concepts of C programming language and problem solving through programming. It covers basic structure of C program, data types, operators, decision making statements, loops, functions, strings, pointers, and also file manipulations. At the end of this course, students will be able to design, implement, test and debug complex problems using features of C.

MODULE-1**UNIT-1****8L+0T+16P=24 Hours****INTRODUCTION TO ALGORITHMS AND PROGRAMMING LANGUAGES:**

Introduction to Algorithms: Basics of algorithms; Flow charts; Generations of programming languages. Introduction to C: Structure of a C program - pre-processor statement, inline comments, variable declaration statements, executable statements; C Tokens - C character set, identifiers and keywords, type qualifiers, type modifiers, variables, constants, punctuations and operators.

Data Types and Operators: Basic data types; Storage classes; Scope of a variable; Formatted I/O; Reading and writing characters; Operators - assignment, arithmetic, relational, logical, bitwise, ternary, address, indirection, sizeof, dot, arrow, parentheses operators; Expressions - operator precedence, associative rules.

Control Statements: Introduction to category of control statements; Conditional branching statements - if, if - else, nested-if, if - else ladder, switch case; Iterative statements - for, while, do - while, nested loops; Jump statements - break, jump, goto and continue.

UNIT-2**8L+0T+16P=24 Hours****ARRAYS & STRINGS:**

Arrays: Introduction; Types of arrays; Single dimensional array - declaration, initialization, usage, reading, writing, accessing, memory representation, operations; Multidimensional arrays.

Strings: Character array, Reading string from the standard input device, Displaying strings on the standard output device, Importance of terminating a string, Standard string library functions.

PRACTICES:**QUESTIONS ON DATA HANDLING – LEVEL 1:**

- Write a program to accept a character as input from the user and print it.
- Write a program to accept a number as input from the user and print it.
- Write a program to accept a float value from the user and print it.
- Write a program to accept a message as input from the user and print it.
- Write a program to accept a message from the user as input and print it in 3 different lines.
- Write a program to accept 2 numbers from the user as input and print their sum.
- Write a program to accept 2 numbers from the user as input and print their product.
- Write a program to accept a number as input from the user which denotes the temperature in Celsius, convert it to Fahrenheit reading and print it.

- Write a program to accept a number as input from the user which denotes the radius and print the area of the circle.
- Write a program to accept a character as input from the user and print its corresponding ASCII value.

QUESTIONS ON CONTROL STATEMENTS - LOOPING – LEVEL 1:

- Write a C program to print all the characters from a to z once.
- Write a C program to print all the characters from Z to A once.
- Write a C program to print all the characters from A to Z 3 times.
- Write a C program to print the first N natural numbers, where N is given as input by the user.
- Write a C program to print the first N natural numbers and their sum, where N is given as input by the user.
- Write a C program to print all the odd numbers between 1 and N where N is given as input by the user.
- Write a C program to print all the even numbers between 1 and N where N is given as input by the user.
- Write a C program to print the squares of the first N natural numbers between 1 and N, where N is given as input by the user.
- Write a C program to print the cubes of the first N natural numbers between 1 and N, where N is given as input by the user.
- Write a C program to print the squares of every 5th number starting from 1 to N, where N is given as input by the user.

QUESTIONS ON CONTROL STATEMENTS – DECISION MAKING – LEVEL 1:

- Write a program to accept two numbers as input check if they are equal.
- Write a program to accept two characters as input and check if they are equal.
- Write a program to accept two numbers as input and print the greater of the 2 numbers.
- Write a program to accept two numbers as input and print the lesser of the 2 numbers.
- Write a program to accept 3 numbers as input and print the maximum of the 3.
- Write a program to accept 3 numbers as input and print the minimum of the 3.
- Write a program to accept a number as input and print EVEN if it is an even number and ODD if it is an odd number.
- Write a program to accept a number as input and check if it is divisible by 3. If it is divisible by 3 print YES else print NO.
- Write a program to accept a number as input and check if it is divisible by both 3 & 5. If it is divisible print YES else print NO.
- Write a program to accept a number as input and check if it is positive, negative or zero.

QUESTIONS ON PATTERNS – LEVEL 1:

1. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.

2. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.

* *
* *
* *

SKILLS:

- ✓ Analysis of the problem to be solved.
- ✓ Select static or dynamic data structures for a given problem and manipulation of data items.
- ✓ Application of various file operations effectively in solving real world problems.
- ✓ Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.

3. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
*
**

4. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
*
**

5. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
1
12
123
1234
12345
6. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
1
22
333
4444
55555
7. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
54321
4321
321
21
1
8. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
12345
2345
345
45
5
9. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
A
AB
ABC
ABCD
ABCDE

10. Write a program to accept a number N as input from the user and print the following pattern.
Sample N = 5.
- ```
A
BC
DEF
GHIJ
KLMNO
```

**QUESTIONS ON NUMBER CRUNCHING – LEVEL 1:**

- Write a program to accept a number as input and print the number of digits in the number.
- Write a program to accept a number as input print the sum of its digits.
- Write a program to accept a number as input, reverse the number and print it.
- Write a program to accept a number and digit as input and find the number of occurrences of the digit in the number.
- Write a program to accept a number as input and check if it is an Armstrong number.
- Write a program to accept a number as input and check if it is an Adam number.
- Write a program to accept a number as input and check if is a prime number.
- Write a program to accept 2 numbers as input and check if they are amicable or not.
- Write a program to accept a number as input and check if it is a power of 2.
- Write a program to accept 2 numbers as input and find their LCM.

**QUESTIONS ON ARRAYS – LEVEL 1:**

- Print the contents of an array from the left to the right.
- Print the contents of an array from the right to the left.
- Find the sum of the elements of an array.
- Find the maximum element in an unsorted array.
- Find the minimum element in an unsorted array.
- Find the average of the elements in an unsorted array.
- Count the number of 0s and 1s in an array having 0s and 1s in random order.
- Count the number of elements in an array whose elements are lesser than a key element in an unsorted array.
- Print all the elements in an array whose values are lesser than a key element in an unsorted array.
- Find the repeated elements in a sorted array.

**QUESTIONS NUMBER CRUNCHING – LEVEL 2:**

- Write a program to accept a number as input and print the product of its digits.
- Write a program to accept a number as input and check if it is a palindrome.
- Write a program to accept a number as input and print the frequency of occurrence of each digit.
- Write a program to accept a number as input and print its factors.
- Write a program to accept a number as input and print its prime factors.
- Write a program to accept a number as input and check if it is a perfect square or not.
- Write a program to accept 2 numbers as input and check if they are betrothed numbers or not.
- Write a program to accept 2 numbers as input and print their HCF.
- Write a program to accept a number as input and check if is a strong number.
- Write a program to generate prime numbers between two intervals given as input.

**QUESTIONS ON ARRAYS – LEVEL 2:**

- Find the sum of the maximum and minimum numbers of an unsorted array.
- Replace every element in an array with the sum of its every other element.
- Replace every element in an array with the sum of its right side elements.
- Replace every element in an array with the sum of its left side elements.
- Reverse the elements of an array (in place replacement).
- Reverse the first half of an array.

- Reverse the second half of an array.
- Write a program to find the second largest element in an unsorted array.
- Write a program to find the second smallest element in an unsorted array.
- Write a program to print the number of odd and even numbers in an unsorted array.

**QUESTIONS ON STRINGS – LEVEL 1:**

- Write a program to accept a string as input and print it.
- Write a program to accept a string as input and count the number of vowels in it.
- Write a program to accept a string as input and count the number of consonants in it.
- Write a program to accept a string as input and print its length.
- Write a program to accept a string as input and print the reversed string.
- Write a program to accept 2 strings as input and check if they are the same.
- Write a program to accept a string as input and copy the contents into a second string and print the second string.
- Write a program to accept 2 strings as input and concatenate them into a third string and print the third string.
- Write a program to accept a string as input and check if it is a palindrome.
- Write a program to accept two strings as input and check if the second string is a substring of the first.

**QUESTIONS ON STRINGS – LEVEL 2:**

- Implement the string length function.
- Implement the string copy function.
- Implement the string concatenate function.
- Implement the string compare function.
- Implement the vowel count function.
- Implement the consonant count function.
- Implement the count words function.
- Implement the string reverse function.
- Implement the strstr function.
- Complete the code snippet to implement the is Palindrome function that checks if a given string is a palindrome. You will need to use the 3 functions string Copy, str Reverse and string Compare functions provided to accomplish this.

**MODULE-2****UNIT-1****8L+0T+16P=24 Hours****FUNCTIONS & POINTERS**

**User-defined functions:** Function declaration - definition, header of a function, body of a function, function invocation; Call by value; Call by address; Passing arrays to functions; Command line arguments; Recursion; Library Functions.

**Pointers:** Declaration, Initialization, Multiple indirection, Pointer arithmetic, Relationship between arrays and pointers, Scaling up - array of arrays, array of pointers, pointer to a pointer and pointer to an array; Dynamic memory allocation functions.

**UNIT- 2****8L+0T+16P=24 Hours****STRUCTURES, UNIONS & FILES**

**Structures:** Defining a structure, Declaring structure variable, Operations on structures, Pointers to structure - declaring pointer to a structure, accessing structure members using pointer; Array of structures, Nested structures, Passing structures to functions - passing each member of a structure as a separate argument, passing structure variable by value, passing structure variable by reference/ address; Typedef and structures.

**Unions:** Defining a union - declaring union variable, operations on union; Pointers to union - declaring pointer to a union, accessing union members using pointer; Array of union, Nested union, Typedef and union, Enumerations, Bit-fields.

**Files:** Introduction to files, Streams, I/O using streams – opening a stream, closing stream; Character input, Character output, File position indicator, End of file and errors, Line input and line output, Formatted I/O, Block input and output, File type, Files and command line arguments.

## PRACTICES:

### QUESTIONS ON STRINGS – LEVEL 3:

- Write a program to swap two given strings and print the swapped strings.
- Write a program to swap two given words of the given sentence and print the altered string.
- Return the maximum occurring character in the string.
- Write a program to print the character in the string with the count where count is the occurrence of the character.
- Write a program to print the duplicate characters in the given string.
- Write a program to remove the duplicate characters in the given string.
- Write a program to remove the vowels from a given string.
- Write a program to rotate a given string N number of times.
- Write a program to check if 2 strings are rotations of each other.
- Write a program to remove the characters from the first string that are present in the second string.

### QUESTIONS ON 2D ARRAYS – LEVEL 1:

- Print the contents of a 2D array row-wise.
- Print the contents of a 2D array column-wise.
- Print the contents of a 2D array in a zig-zag order.
- Print the contents of a 2D array diagonal-wise.
- Print the contents of a 2D array right-diagonal order.
- Print the contents of a 2D array left-diagonal order.
- Print the contents of a 2D array in the upper triangular order – left top to right bottom.
- Print the contents of a 2D array in the lower triangular order.
- Find and print the maximum element along with its position in a matrix.
- Find and print the minimum element along with its position in a matrix.

### QUESTIONS ON 2D ARRAYS – LEVEL 2:

- Find and print the maximum element of each row of a matrix.
- Find and print the minimum elements of each row of a matrix.
- Find and print the maximum element of each column of a matrix.
- Find and print the minimum element of each column of a matrix.
- Find the lowest value in the upper triangle area and the largest value in the lower triangular area of a matrix and print their product.
- Find the sum of the elements of each row and each column of a matrix and print the minimum row sum and maximum sum column.
- Write a program to find the row with the maximum number of 1's in a matrix consisting of only 0's and 1's.
- Write a program to print the quotient and remainder on dividing sum of left-top to right-bottom diagonal by sum of right-top to left-bottom diagonal.
- Write a program to print the absolute difference of the sum of major diagonal elements and the sum of minor diagonals of the given matrix.
- Write a program to search a given element in a row-wise and column-wise sorted 2D array.

**QUESTIONS ON 2D ARRAYS – LEVEL 3:**

- Write a program to find the Kth smallest element in the given matrix.
- Write a program to find the Kth largest element in the given matrix.
- Write a program to check whether the given two two-dimensional array of same dimensions are equal or not.
- Write a program to add the given two two-dimensional array of same dimensions.
- Write a program to subtract the given two two-dimensional array of same dimensions.
- Write a program to multiply the given two two-dimensional array of same dimensions.
- Write a program to sort each row of a matrix.
- Write a program to find the sum of the elements in 'Z' sequence of the given 2D array.
- Write a program to print the unique rows of the given two-dimensional array consisting of only 0's and 1's.
- Write a program to print the unique columns of the given two-dimensional array consisting of only 0's and 1's.

**QUESTIONS ON FILES, STRUCTURES & UNIONS:**

1. Write a C program to create a struct, named Student, representing the student's details as follows: first\_name, last\_name, Age and standard.

**Example**

Read student data

john

carmack

15

10

Display the data in the following format

First Name: john

Last Name: carmack

Age: 15

Standard: 10

2. Declare a structure POINT. Input the coordinates of point variable and write a C program to determine the quadrant in which it lies. The following table can be used to determine the quadrant.

| Quadrant | X        | Y        |
|----------|----------|----------|
| 1        | Positive | Positive |
| 2        | Negative | Positive |
| 3        | Negative | Negative |
| 4        | Positive | Negative |

**Example**

Input the values for X and Y coordinate: 7 9

The coordinate point (7,9) lies in the First quadrant.

3. Bob and Alice both are friends. Bob asked Alice how to store the information of the books using Structures. Then Alice written a c program to store the information of books using book structure by taking different attributes like book\_name, author, book\_id, price. Write a C program to read and display the attributes of the books using structures.

**Sample Input:**

Enter number of books: 1

Enter the book name: c Programming

Enter the author name: balaguruswamy

Enter the book ID: 23413

Enter the book price: 500

**Sample Output:**

The details of the book are:

The book name is: c Programming

The author name is: balaguruswamy

The book ID is: 23413

The book price is: 500.00

4. Ramesh wants to do addition on complex numbers. He did it with regular practice but Charan asked him to do with the help of structures by following below Criteria.

Write a C program that defines a structure named 'Complex' consisting of two floating point members called "real and imaginary". Let c1 and c2 are two Complex variables; compute the sum of two variables.

**Example:**

c1= 2 8

c2= 6 4

Sum= 8.000000+12.000000i

5. Customer Payment Details is a structure with members as customers\_name, address, account\_number, payment\_status(paid(1)/ not\_paid(0)), due\_date, and amount. In this example, payment\_date is another structure with month, day and year as integer members. So, every customer record can be considered as an array of structures.

Write a C program that displays the amount to be paid by each customer along with their names. If payment\_status is 1, display NIL for such customers.

**Input Format:**

First line of input contains 'n' number of customers, followed by 8 lines of input for each customer. Each line represents (customers\_name, address, account\_number, amount payment\_status(paid(1)/ not\_paid(0)), and due\_date).

**Output Format:** First line of output is Amount to be paid by each customer as on date: followed by n lines of output. Each line contains name of the customer followed by tab space, and amount to be paid.

**Hint:** Use nested structure to represent date.

6. Write a 'C' program to accept customer details such as: Account\_no, Name, Balance using structure. Assume 3 customers in the bank. Write a function to print the account no. and name of each customer whose balance < 100 Rs.
7. Write a C program to accept details of 'n' employee(eno, ename, salary) and display the details of employee having highest salary. Use array of structure.
8. Write a C program to print the bill details of 'N' number of customers with the following data: meter number, customer name, no of units consumed, bill date, last date to deposit and city. The bill is to be calculated according to the following conditions:

| No. of units           | Charges          |
|------------------------|------------------|
| For first 100 units    | Rs.0.75 per unit |
| For the next 200 units | Rs.1.80 per unit |
| For the next 200 units | Rs.2.75 per unit |

**Sample Input**

Enter no. of customers

1

Enter Meter Number AP01213

Enter Customer Name: Karthik  
 Enter No. of units consumed: 200  
 Enter Bill date:22/01/2021  
 Enter Last date: 12/2/2021  
 Enter City: Guntur

Sample Output

Meter Number AP01213  
 Customer Name: Karthik  
 No. of units consumed: 200  
 Bill date:22/01/2021  
 Last date: 12/2/2021  
 City: Guntur  
 Total Amount: 255.000000

9. Write a C program that creates a student file containing {Roll No, Student Name, Address, Stream}, where the data will be inserted and display the list of students who are in CSE (Stream=CSE).

**Input:** A file name

**Output:** The attributes such as Roll\_No, Student\_Name, Stream, Address.

**Sample Input**

|           |      |     |        |
|-----------|------|-----|--------|
| 201fa4200 | Raja | CSE | Guntur |
| 201fa4201 | Bala | IT  | Tenali |

**Sample Output**

|           |      |     |        |
|-----------|------|-----|--------|
| 201fa4200 | Raja | CSE | Guntur |
|-----------|------|-----|--------|

10. Write a C program that reads content from an existing text file and write the same in a new file by changing all lowercase alphabetic character to upper case. (Existing file may contain digit and special characters).

**Example:**

**Input:** Enter the file name.

**Output:** New file with updated content.

11. Write a C program to count the occurrences of the given string in a file.

**Example:**

Input: Enter the File name to read the string to be counted.

Output: Display the count of occurrences of the string.

12. Write a C Program to transfer the data from one location to another location without changing the order of the content.

**Example:**

Read the file name from the user. If the source file exists, Transfer the data and display the message as "Data is transferred successfully" otherwise display the message "No such file is existing in the directory."

13. Write a C program that reads numbers and write them into a text-file. Also find odd and even numbers in that file and store it in 2 separate files named odd.txt and even.txt. All the values should be in ascending order.

**Input:** Enter the values.

**Output:** Creates a separate file for Even and Odd numbers.

**Sample Input:**

4 43 2 53 45

**Sample Output:**

Even.txt: 2 4

Odd.txt: 43 45 53

14. Write a C program to replace the content in the given text file.

**Input:** Enter the file name, line number to be replaced and the new content

**Output:** New file with replaced lines.

**Example:****Sample Input:**

Enter the file name: abc.txt

Enter the line no to replace: 3

Enter the content: Files stores data presently.

**Sample Output:**

Line no 3 is replaced with the given content.

The content of the file abc.txt contains:

test line 1

test line 2

Files stores data presently

test line 4

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

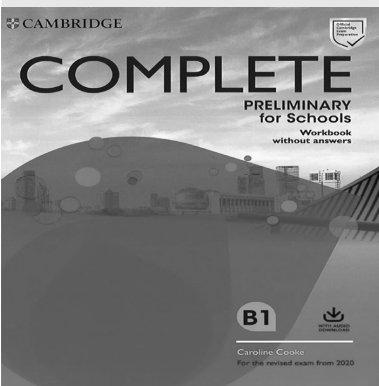
| CO No. | Course Outcomes                                                                                                                                                  | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Identify suitable data type for operands and design of expressions having right precedence.                                                                      | Apply        | 1, 2       | 1                |
| 2      | Apply decision making and iterative features of C Programming language effectively.                                                                              | Apply        | 1, 2       | 1                |
| 3      | Select problem specific data structures and suitable accessing methods.                                                                                          | Analyze      | 1, 2       | 1, 2             |
| 4      | Design and develop non- recursive and recursive functions and their usage to build large modular programs and also able to design string manipulation functions. | Create       | 1, 2       | 3                |
| 5      | Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.                                     | Evaluate     | 1, 2       | 3, 4             |

**TEXT BOOKS:**

- Behrouz A. Forouzan, Richard F.Gilberg, "Programming for Problem Solving", 1st edition, Cengage publications, 2019.
- Ajay Mittal, "Programming in C - A Practical Approach", 1st edition, Pearson Education, India, 2010.

**REFERENCE BOOKS:**

- Reema Thareja, "Computer Fundamentals and Programming in C", 1st edition, Oxford University Press, India, 2013.
- Herbert Schildt, "C: The Complete Reference", 4th edition, Tata McGraw-Hill, 2017.
- Byron S Gottfried, "Programming with C", 4th edition, Tata McGraw-Hill, 2018.



Source: <https://www.scribd.com/document/502301821/Cambridge-Complete-B1-Preliminary-for-Schools-Workbook-2020-Edition>

## 22EN102 ENGLISH PROFICIENCY AND COMMUNICATION SKILLS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 2 | 1 |

**PREREQUISITE KNOWLEDGE:** Basics of grammar, Read and understand for global context, Cultural sensitivity and Basic writing skills.

### COURSE DESCRIPTION AND OBJECTIVES:

English Proficiency and Communication Skills seeks to develop the students' abilities in grammar, speaking, reading, writing and overall comprehension skills. The course will provide students an exposure on a wide range of language use in everyday situations. It will make the students to equip with functional English and make them use it confidently in their professional and social contexts. Finally, students will strengthen their reading, writing, listening and speaking skills in English

### MODULE-1

#### UNIT-1

0L+0T+8P=8 Hours

#### MY LIFE AND HOME - MAKING CHOICES - HAVING FUN:

**Reading:** Understanding main message, factual information global meaning, specific information and paraphrasing.

**Writing:** Developing hints based mail, Writing short messages/paragraphs.

**Listening:** Understanding short monologues or dialogues and choose the correct visual.

**Speaking:** Express simple opinions /cultural matters in a limited way.

**Vocabulary:** Discerning use of right word suiting the context, B1 Preliminary word list.

**Grammar:** Frequency Adverbs, State Verbs, AFV and Prepositions.

#### UNIT-2

0L+0T+8P=8 Hours

#### ON HOLIDAY - DIFFERENT FEELINGS – THAT'S ENTERTAINMENT!:

**Reading:** Longer text for detailed comprehension, gist and inference.

**Writing:** Developing notes and responding to penfriends or 'e-pals'.

**Listening:** Understand straightforward instructions or public announcements.

**Speaking:** Describing people, things and places in a photograph.

**Vocabulary/Grammar:** Comparatives and Superlatives, Gradable and non-gradable adjectives, Cloze tests.

#### PRACTICES:

- Developing hints based mail.
- Writing short message.
- Writing paragraphs.
- Expressing opinions and cultural matters.
- Understanding short monologues.
- Understanding straightforward instructions and public announcements.
- Describing people, things and places in a photograph.



**MODULE-2****UNIT-1****0L+0T+8P=8 Hours****GETTING AROUND – INFLUENCES - STAY FIT AND HEALTHY:****Reading:** Reading for understanding coherence of the text and drawing inferences.**Writing:** Reading an announcement from a magazine or website for preparing an article.**Listening:** Discussion activities and listening to understand the gist of each short dialogue.**Speaking:** Snap Talks, Make and respond to suggestions, discuss alternatives and negotiate agreement.**Vocabulary / Grammar:** Punctuation, Prepositions, Phrasal Verbs, B1 Preliminary word list.**UNIT-2****0L+0T+8P=8 Hours****LOOKS AMAZING! - THE NATURAL WORLD - EXPRESS YOURSELF!:****Reading:** Content, Communicative Achievement, Organisation and Language.**Writing:** Developing a story with clear links to the given opening sentence.**Listening:** An interview for a detailed understanding of meaning and to identify attitudes and opinions.**Speaking:** Discuss likes, dislikes, experiences, opinions, habits, etc.**Vocabulary/Grammar:** Modals, Conditionals, Verb forms (Time and Tense).**PRACTICES:**

- Listening to understand the gist of each short dialogue.
- Listening to an interview for a detailed understanding of meaning and to identify attitudes and opinions.
- Preparing an article.
- Discuss for alternatives and negotiate agreement.
- Discussion on likes, dislikes, experiences, opinions, habits, etc.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                         | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply to read and grasp content on a range of topics/texts related to their everyday life like notifications, advertisements, travel brochures, news reports, articles. | Apply        | 1          | 7, 8, 9, 10, 12  |
| 2      | Apply suitable strategies to achieve comprehension, like listening for main points and checking comprehension using contextual clues etc.                               | Apply        | 1          | 7, 8, 9, 10, 12  |
| 3      | Use functional English to communicate and interact effectively in everyday situations.                                                                                  | Apply        | 1, 2       | 7, 8, 9, 10, 12  |
| 4      | Demonstrate vocabulary beyond that of the familiar subjects.                                                                                                            | Analyze      | 1, 2       | 7, 8, 9, 10, 12  |
| 5      | Show sufficient control of English grammar and sentence variety to coherently organise information at sentence and discourse levels.                                    | Evaluate     | 2          | 7, 8, 9, 10, 12  |

**TEXT BOOK:**

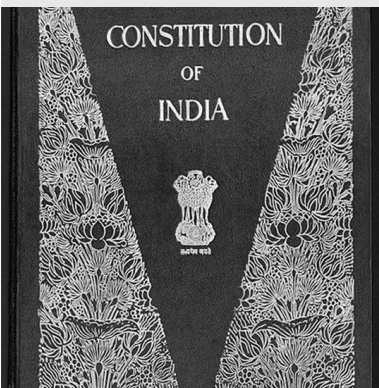
1. Emma Heyderman and Peter May, "Complete Preliminary", Student's Book with Answers, 2nd edition, Cambridge University Press, 2019.

**REFERENCE BOOKS:**

1. Annette Capel and Rosemary Nixon, "Introduction to PET", Oxford University Press, 2009.
2. Adrian Doff and Craig Thaine, "Empower Pre intermediate", Cambridge University Press, 2015.
3. Louise Hashemi and Barbara Thomas, "Objective PET", Cambridge University Press, 2010.

**SKILLS:**

- ✓ Use of appropriate grammar and vocabulary with syntactic patterns in short texts.
- ✓ Read and extract the main message, global meaning, specific information, detailed comprehension, understanding of attitude, opinion and writer purpose and inference.
- ✓ Listen to understand key information, specific information, gist and detailed meaning and to interpret meaning.
- ✓ Understand questions and make appropriate responses and talk freely on everyday topics



Source: [https://commons.wikimedia.org/wiki/File:Constitution\\_india.jpg](https://commons.wikimedia.org/wiki/File:Constitution_india.jpg)

## 22TP101 CONSTITUTION OF INDIA

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 0 | 2 | 0 | 1 |

**PREREQUISITE KNOWLEDGE:** High School-level Civics and Social Studies.

### COURSE DESCRIPTION AND OBJECTIVES:

To provide students with a basic understanding of Indian Polity and Constitution and make students understand the functioning of government at the center and state level besides local self-government. This course also equips students with knowledge pertaining to fundamental rights and fundamental duties of a citizen in a democracy such as India.

### MODULE-1

#### UNIT-1

**0L+8T+0P=8 Hours**

#### HISTORICAL BACKGROUND TO THE INDIAN CONSTITUTION:

Meaning of the constitution law and constitutionalism; Historical perspective of the Constitution of India; Salient features and characteristics of the Constitution of India.

#### UNIT-2

**0L+8T+0P=8 Hours**

#### FUNDAMENTAL RIGHTS, DUTIES, DIRECTIVE PRINCIPLES, AND AMENDMENT:

Scheme of the fundamental rights - scheme of the Fundamental Right to Equality; scheme of the Fundamental Right to certain Freedom under Article 19; scope of the Right to Life and Personal Liberty under Article 21; Scheme of the Fundamental Duties and its legal status; Directive Principles of State Policy – its importance and implementation; Amendment of the Constitution - Powers and Procedure.

#### PRACTICES:

- Enactment of Constituent Assembly debates to further understand the rationale for the provisions of the constitution.
- Fundamental Rights in our popular culture - discussion in the movie Jai Bhim.

### MODULE-2

#### UNIT-1

**0L+8T+0P=8 H Hours**

#### STRUCTURE AND FORM OF GOVERNMENT:

Federal structure and distribution of legislative and financial powers between the Union and the States; Parliamentary Form of Government in India – The constitution powers and status of the President of India; Emergency Provisions: National Emergency, President Rule, Financial Emergency.

#### UNIT-2

**0L+8T+0P=8 Hours**

#### LOCAL SELF GOVERNMENT:

Local Self Government – Constitutional Scheme in India – 73rd and 74th Amendments.

#### PRACTICES:

- Debate on federalism in India.
- Collect news published in the local papers about panchayats in the nearby areas.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                       | Blooms Level | Module No. | Mapping with POs |
|--------|-----------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Analyse major articles and provisions of the Indian constitution.     | Analyze      | 1          | 6                |
| 2      | Appreciation for the constitution and safeguarding individual rights. | Apply        | 1          | 6                |
| 3      | Evaluating functions of various organs of the State in a democracy.   | Evaluate     | 2          | 6                |

**TEXT BOOK:**

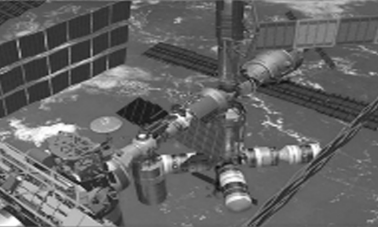
1. PM Bhakshi, "Constitution of India", 15th edition, Universal Law Publishing, 2018.

**REFERENCE BOOKS:**

1. B. R. Ambedkar, "The Constitution of India" Educreation Publishing, India, 2020.
2. Subhash Kashyap, "Our Constitution" 2nd edition, National Book Trust, India, 2011.
3. Arun K. Thiruvengadam, "The Constitution of India: A Contextual Analysis", Hart Publishing India, 2017.

**SKILLS:**

- ✓ *Knowledge the basics of the Indian constitution.*
- ✓ *Know the fundamental rights, fundamental duties, and Directive Principles of State Policy.*
- ✓ *Fair knowledge about the functioning of various institutions in a democracy*



Source: <https://www.elsevier.com/books/advanced-engineering-mathematics/jeffrey/978-0-08-052296-8>

# 22MT105 ADVANCED ENGINEERING MATHEMATICS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 4 |

**PREREQUISITE KNOWLEDGE:** Differentiation, Integration, Vectors.

## COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is students should be able to adapt their knowledge and techniques to solve other similar application problems. Make aware students about the importance and symbiosis between Mathematics and Engineering. To develop the mathematical skill so that students are able to apply mathematical methods & principals in solving problem from Engineering fields. Students will learn applications of differentiation, integration and vectors to use in solving problem from Engineering fields.

## MODULE-1

### UNIT-1

12L+8T+0P=20 Hours

#### PARTIAL DIFFERENTIATION:

Partial differentiation, Homogeneous functions, Euler's theorem, Total differential coefficient.

### UNIT-2

12L+8T+0P=20 Hours

#### APPLICATIONS OF PARTIAL DIFFERENTIATION:

Maxima and Minima of a function of two variables, Conditions for extreme values, Lagrange method of undetermined multipliers, Jacobians.

#### PRACTICES:

- Write basic formulas of differentiation and integration
- Finding the differentiation and integration of functions of several variables.
- What is the tangent and normal?
- Find the velocity and acceleration of given function.
- Tell about maxima and minima.
- How to know given function is increasing or decreasing?

## MODULE-2

### UNIT-1

12L+8T+0P=20 Hours

#### VECTOR CALCULUS:

Introduction to vectors, Vector algebra (Review), Scalar and vector point functions, Gradient, Divergence and Curl.

Introduction to Multiple integrals (Review), Line integral, Surface integral, Volume integral.

### UNIT-2

12L+8T+0P=20 Hours

#### APPLICATIONS OF VECTOR CALCULUS:

Normal vector, Directional Derivate, Solenoidal and Irrotational vectors, Scalar potential.

Green's theorem for plane, Gauss divergence theorem, Stokes' theorem (without proofs).

**PRACTICES:**

- Tell difference between a vector and scalar.
- Algebra of a vectors.
- Find the area and volume bounded by the curves
- Differentiate between line, surface and volume integrals
- Applications of vector differential operators
- What is tangent and normal vector?

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                        | Blooms Level | Module No. | Mapping with POs   |
|--------|------------------------------------------------------------------------|--------------|------------|--------------------|
| 1      | Apply differentiation and integration for solving engineering problems | Apply        | 1          | 1, 2, 4, 9, 10, 12 |
| 2      | Apply vector differential operators in real life problems              | Apply        | 2          | 1, 2, 4, 9, 10, 12 |
| 3      | Analyse a function for its extreme values                              | Analyze      | 1          | 1, 2, 4, 9, 10, 12 |
| 4      | Evaluate line, surface and volume integrals                            | Evaluate     | 2          | 1, 2, 4, 9, 10, 12 |

**TEXT BOOKS:**

1. N. P. Bali, K. L. Sai Prasad, "A Textbook of Engineering Mathematics I, II, III", 2nd Edition, Universal Science Press, New Delhi, 2018.
2. B. S. Grewal, "Higher Engineering Mathematics", 44th Edition, Khanna Publishers, 2018.

**REFERENCE BOOKS:**

1. H. K. Dass and Er. Rajanish Verma, "Higher Engineering Mathematics", Third edition S. Chand and Co., 2015.
2. B. V. Ramana, "Advanced Engineering Mathematics", TMH Publishers, 2020.
3. T. K.V. Iyengar et al, "Engineering Mathematics, I, II, III", S. Chand and Co., New Delhi, 2018.

**SKILLS:**

- ✓ Needs to evaluate integration and differentiation of all functions.
- ✓ Understand how to choose the limits from the given curves.
- ✓ Need to understand the geometrical representation of each given curve.

# 22MT107 DISCRETE MATHEMATICAL STRUCTURES

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 0 | 3 |

**PREREQUISITE KNOWLEDGE:** Basic of set theory, Algebra.

## COURSE DESCRIPTION AND OBJECTIVES:

The course objective is to provide students with an overview of discrete mathematics. Students will learn about of group and expected to demonstrate analytical and combinatorial methods such as propositional logic , Mathematical Induction, Boolean functions, combinatorial ,recurrence relation, generating function and graph theory

## MODULE-1

### UNIT-1

8L+8T+0P=16 Hours

#### BOOLEAN ALGEBRA AND LOGIC:

**Group:** Group, Subgroup, Lagrange's theorem, Introduction to rings and fields.

**Boolean algebra:** Boolean algebra, Truth table, Basic logic gates, Postulates of Boolean algebra, Principle of duality, Propositions, Connectives, Equivalence and Normal form.

### UNIT-2

8L+8T+0P=16 Hours

#### APPLICATIONS OF LOGIC & BOOLEAN ALGEBRA:

**Normal form:** CNF, DNF, CDNF, CCNF, Conversion of CNF to DNF and vice versa. Minimization of Boolean function (Karnaugh Maps), Generalized Pigeonhole Principle.

#### PRACTICES:

- Determine order of an element of group.
- Determine the order of a subgroup of a finite group.
- Determine whether algebraic structure is a Ring or a Field.
- Construct Truth table of propositions.
- Check whether propositions are equivalence.
- Obtain CNF, DNF of expression.
- Draw 2 – variables, 3-variables K-map.
- Minimize the Boolean function by K-map.
- Simplify the Boolean expression using Boolean algebra laws.
- Determine the homogeneous solution and particular solution for recurrence relation.
- Switching Circuit in Boolean algebra, Combination of two switches in a Circuit

## MODULE-2

### UNIT-1

8L+8T+0P=16 Hours

#### COMBINATORICS AND GRAPHS:

**Combinatorics:** The basics of counting, Permutations and combinations, Discrete Numeric Functions. Recurrence relations and Generating functions.

**Graph theory:** Graph terminology, Special types of graphs, Connected graph, Weighed graph, Graph Isomorphism, Euler and Hamiltonian paths and circuits, Planar graphs, Bipartite graph, Tree.

## DISCRETE MATHEMATICAL STRUCTURES

Sixth Edition

For Third Semester B.E. Classes  
(CSE and ISE Branches)

As per Revised VTU Syllabus 2019-20



Dr. D.S.C

PRISM

Source: <https://www.amazon.in/Discrete-Mathematical-Structures-Dr-D-S-C/dp/9388478398>

**UNIT-2****8L+8T+0P=16 Hours****APPLICATIONS OF COMBINATORICS AND GRAPHS:**

Graph Coloring, Chromatic number, Matrix representation of graph, DFS, BFS algorithms, Minimum spanning tree.

**PRACTICES:**

- Represent sequence by Generating function.
- Solve Generation function by recurrence relation.
- Find degree of digraph and undirected graph.
- Identify Isomorphic graph, Euler circuit, Hamiltonian circuit.
- Determine matrix representation of graph.
- Determine the regions of planer connected graph.
- Draw  $K_n$ ,  $K_{m,n}$ .
- Determine the chromatic number of graphs.
- Find the weight of minimum spanning tree.
- Find number of arrangements that are possible.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                               | Blooms Level | Module No. | Mapping with POs |
|--------|---------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply the concepts of mathematical logic and Boolean algebra. | Apply        | 1          | 1, 2, 9, 10, 12  |
| 2      | Apply Karnaugh map to minimize.                               | Apply        | 1          | 1, 2, 9, 10, 12  |
| 3      | Solve generating function by recurrence relations.            | Apply        | 2          | 1, 2, 9, 10, 12  |
| 4      | Model and solve real world problems using graphs and trees.   | Apply        | 2          | 1, 2, 9, 10, 12  |

**TEXT BOOKS:**

1. Tremblay, J.P. and Manohar. R, "Discrete Mathematical Structures with Applications to Computer Science", 30th Reprint, Tata McGraw Hill Pub. Co. Ltd, New Delhi, 2017.
2. Rosen, K.H., "Discrete Mathematics and its Applications", 7th Edition, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 2018.

**REFERENCE BOOKS:**

1. R.P. Grimaldi, "Discrete and Combinatorial Mathematics: An Applied Introduction", 4th Edition, Pearson Education Asia, Delhi, 2017.
2. S. Lipschutz and Mark Lipson., "Discrete Mathematics", Schaum's Outlines, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 2021.
3. T. Koshy, "Discrete Mathematics with Applications", Elsevier Publications, 2015.
4. S B Singh. "Discrete Structures", Khanna Book Publishers Co-Pvt. Ltd. 2019.

**SKILLS:**

- ✓ Familiarity of concepts of statements, logic and truth tables.
- ✓ Analyze closed form of discrete numeric function.
- ✓ Know some basic properties of graphs, trees and related discrete structures.



# 22ME101 ENGINEERING GRAPHICS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 2 | 3 |

**PREREQUISITE KNOWLEDGE:** Basics of Geometry.

## COURSE DESCRIPTION AND OBJECTIVES:

Engineering graphics is the language of engineers and is the most effective way of communicating and sharing technical ideas in the form of pictures/drawings. The objective of this course is to familiarize the students with the conventional concepts of engineering drawing and computer aided drawing.

## MODULE-1

### UNIT-1

6L+0T+6P=12 Hours

#### ENGINEERING CURVES:

Types of lines; Lettering, Dimensioning, Geometric constructions - lines, polygons (Angle, ARC, General and Inscribe in circle method), Conical curves (General method), Ellipse by Oblong method.

### UNIT-2

10L+0T+10P=20 Hours

#### ORTHOGRAPHIC PROJECTIONS OF POINTS, LINES & PLANES:

Principles of projection; Projections of points; Projection of straight lines - Inclined to one plane, inclined to both planes; Projection of planes - Inclined to one plane.

#### PRACTICES:

- Construction of polygons using different methods (i.e. ARC, Angle, General).
- Inscribe a regular hexagon & pentagon in a circle of the given diameter.
- Tracing of conical curves (Ellipse, Parabola, Hyperbola) by using General Method.
- Draw the projections of the points situated in all the 4 quadrants.
- Draw the projections of a line when it is inclined to one plane (HP or VP).
- Draw the projections of a line when it is inclined to both the planes (HP & VP).
- Draw the projections of a plane when it is inclined to one plane (HP or VP).

## MODULE-2

### UNIT-1

6L+0T+6P=12 Hours

#### PROJECTIONS OF SOLIDS:

Projection of solids axis inclined to one reference plane - Prisms, pyramids, Cylinder and cone.

#### DEVELOPMENT OF SURFACES:

Development of lateral surfaces of simple solids - Prisms, Pyramids, Cylinder and cone.

### UNIT-2

10L+0T+10P=20 Hours

#### ORTHOGRAPHIC VIEWS:

Conversion of pictorial views into orthographic views.

**Drafting Using Computer Package:** Introduction to 2D modelling software - AutoCAD; Conversion of Isometric view into Orthographic views of simple castings; Conversion of Orthographic views into Isometric view of simple solids - Prisms, Pyramids, Cylinders and cones.

Source: <https://depositphotos.com/5087383/stock-photo-the-engineering-drawing.html>



**PRACTICES:**

- Draw the projections of Prisms, when they are inclined to one reference plane (HP or VP).
- Draw the projections of Pyramids, when they are inclined to one reference plane (HP or VP).
- Draw the projections of cylinder & cone, when they are inclined to one reference plane (HP or VP).
- Draw the complete surface development of prisms & pyramids with the given dimensions.
- Draw the complete surface development of cylinder & cone with the given dimensions.
- Draw the orthographic view's (i. e. front view, top view, and side view) of the given pictorial view of the sketches by using AutoCAD.
- Draw the Isometric view of simple solids (Prisms & Pyramids) by using AutoCAD.
- Draw the Isometric view of simple solids (Cylinder & Cone) by using AutoCAD.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                          | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Communicate the technical ideas in the form of drawings.                 | Apply        | 1          | 1, 2, 3, 5       |
| 2      | Apply the drawing skills in representing various geometrical features.   | Apply        | 1          | 1, 2, 3, 5       |
| 3      | Develop orthographic projections and isometric views of various objects. | Apply        | 1          | 1, 2, 3, 5       |
| 4      | Estimate the lateral surface area of regular geometrical solids.         | Analyze      | 2          | 1, 2, 3, 5       |
| 5      | Sketch simple objects and their pictorial views using AutoCAD.           | Analyze      | 2          | 1, 2, 3, 5       |

**TEXT BOOKS:**

1. J Hole, "Engineering Drawing", Tata McGraw-Hill, 2nd Edition, 2019.
2. N D Bhatt, "Engineering Drawing", Charotar Publication, 53rd Edition, 2014.

**REFERENCE BOOKS:**

1. Basant Agrawal and C.M. Agrawal "Engineering Drawing", , Tata Mc Graw- Hill, 2nd Edition 2018.
2. K L Narayana, "Engineering drawing", SciTech Publications, 3rd Edition, 2011.
3. Colin H. Simmons, Dennis E. Maguire, Manual of Engineering Drawing, 2nd Edition, 2003.

**SKILLS:**

- ✓ Convert isometric views of objects into orthographic views and vice versa.
- ✓ Visualize the shape of the 3D components.
- ✓ Create pictorial views by using AutoCAD.
- ✓ Know projections by visualization.

# 22TP104 BASIC CODING COMPETENCY

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 0 | 1 | 3 | 2 |

**PREREQUISITE KNOWLEDGE:** Programming in C.

## COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed to impart knowledge on advanced concepts of C programming language and problem solving. At the end of this course, students will be able to design, implement, test and debug complex problems using features of C.

## MODULE-1

### UNIT-1

**0L+4T+12P=16 Hours**

### NUMBER CRUNCHING:

### PRACTICES:

Problems On Number Crunching

- Write a program to check if a given number is perfect or not.
- Write a program to check if a given number is deficient or not.
- Write a program to check if 2 given numbers are amicable or not.
- Write a program to check if 2 given numbers are betrothed or not.
- Write a program to check whether a given number is an Armstrong number or not.
- Write a program to print the series of prime numbers in the given range.
- Write a program to print all the perfect numbers in a given range.
- Write a program to generate all deficient numbers in a given range.
- Write a program to generate all the amicable numbers in a given range.
- Write a program to generate all the betrothed numbers in a given range.
- Write a program to find the largest prime factor of a given number.
- Write a program to check whether the given number is a palindrome or not.
- Write a program to calculate sum of the individual digits for the given number.
- Write a program to find the first number that has more than 'n' factors, excluding 1 and that number.
- Write a program to accept a number as input and print its factorial.
- Write a program to accept a number n, print first N Fibonacci numbers.
- Write a program to check if an input number is Armstrong number or not.
- Write a program that takes input a,b. Print a power b.
- Write a program that takes input a number n, check if it a perfect square or not.
- Print array in spiral format.
- Print sum of each row in a matrix.
- Print sum of each column in matrix.
- Print left->right and right->left diagonals in a matrix.
- Initially you are at (0,0) find the shortest path count to reach the (n, n) block in matrix.
- Remove all the elements present in row and column of unsafe elements. An element is called unsafe if it is equal to smallest or largest value. Count number of remaining elements.
- Write a program to check if the string contains all the letters of alphabet.
- Check if a string is matching password requirements.

COMPETITIVE  
PROGRAMMING

Source: [www.geeksforgeeks.org/best-way-to-start-with-competitive-programming-geeksforgeeks-cp-live-course/](http://www.geeksforgeeks.org/best-way-to-start-with-competitive-programming-geeksforgeeks-cp-live-course/)

- Check if String A contains String B (String searching).
- Check if a number is harshad number or not.
- Write a program to get 3 numbers as input. The first is the number num1 and second is the digit that needs to be replaced. The third is the digit that is to replace the 2nd digit. Print the number after performing this operation.
- Write a program to accept a number and swap its alternate digits. Print the number generated.
- Write a program to accept a number and choice as input. If the choice is 0 rearrange the number such that the odd digits are ordered first followed by the even digits. If the choice is 1 rearrange the number such that the even digits are ordered first followed by the odd digits. Print the rearranged number. The order of occurrence of the digits is to be preserved.
- Write a program to determine that whether the given quadrilateral is cyclic or not. You are given the sizes of angles of a simple quadrilateral (in degrees) A, B, C and D, in some order along its perimeter.

**Note:** A quadrilateral is cyclic if and only if the sum of opposite angles is  $180^\circ$ .

- Chef is a very lazy person. Whatever work is supposed to be finished in x units of time, he finishes it in max units of time. But there is always a limit to laziness, so he delays the work by at max d units of time. Given x,m,d, find the maximum time taken by Chef to complete the work.
- Suppose Chef is stuck on an island and currently he has x units of food supply and y units of water supply in total that he could collect from the island. He needs xr units of food supply and yr units of water supply per day at the minimal to have sufficient energy to build a boat from the woods and also to live for another day. Assuming it takes exactly D days to build the boat and reach the shore, tell whether Chef has the sufficient amount of supplies to be able to reach the shore by building the boat? Read five integers x,y,xr,yr,D from the user and display "YES" if Chef can reach the shore by building the boat and "NO" if not (without quotes).
- There are 3 problems in a contest namely A,B,C respectively. Alice bets Bob that problem C is the hardest while Bob says that problem B will be the hardest.

You are given three integers SA,SB,SC which denotes the number of successful submissions of the problems A,B,C respectively. It is guaranteed that each problem has a different number of submissions. Determine who wins the bet.

- 1) If Alice wins the bet (i.e. problem C is the hardest), then output Alice.
- 2) If Bob wins the bet (i.e. problem B is the hardest), then output Bob.
- 3) If no one wins the bet (i.e. problem A is the hardest), then output Draw.

**Note:** The hardest problem is the problem with the least number of successful submissions.

#### Input Format

- The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first and only line of each test case contains three space-separated integers SA,SB,SC, denoting the number of successful submissions of problems A,B,C respectively.

#### Output Format

For each test case, output the winner of the bet or print Draw in case no one wins the bet.

#### Sample Input 1

```
3
1 4 2
16 8 10
14 15 9
```

#### Sample Output 1

```
Draw
Bob
Alice
```

- In a season, each player has three statistics: runs, wickets, and catches. Given the season stats of two players A and B, denoted by R, W, and C respectively, the person who is better

#### SKILLS:

- ✓ Analysis of the problem to be solved.
- ✓ Application of various file operations effectively in solving real world problems.
- ✓ Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.

than the other in the most statistics is regarded as the better overall player. Tell who is better amongst A and B. It is known that in each statistic, the players have different values.

#### Input

The first line contains an integer T, the number of test cases. Then the test cases follow.

Each test case contains two lines of input.

The first line contains three integers R1, W1, C1, the stats for player A.

The second line contains three integers R2, W2, C2, the stats for player B.

#### Output

For each test case, output in a single line "A" (without quotes) if player A is better than player B and "B" (without quotes) otherwise.

- Write a program to find the direction.

Chef is currently facing the north direction. Each second he rotates exactly 90 degrees in clockwise direction. Find the direction in which Chef is facing after exactly X seconds.

**Note:** There are only 4 directions: North, East, South, West (in clockwise order). Initially chef is at 0th second i.e., facing North direction.

#### Input Format

- First line will contain T, number of testcases. Then the testcases follow.
- Each testcase contains of a single integer X.

#### Output Format

For each testcase, output the direction in which Chef is facing after exactly X seconds.

#### Sample Input 1

```
3
1
3
6
```

#### Sample Output 1

```
East
West
South
```

- Chef is playing in a T20 cricket match. In a match, Team A plays for 20 overs. In a single over, the team gets to play 6 times, and in each of these 6 tries, they can score a maximum of 6 runs. After Team A's 20 overs are finished, Team B similarly plays for 20 overs and tries to get a higher total score than the first team. The team with the higher total score at the end wins the match. Chef is in Team B. Team A has already played their 20 overs, and have gotten a score of R. Chef's Team B has started playing, and have already scored C runs in the first O overs. In the remaining 20-O overs, find whether it is possible for Chef's Team B to get a score high enough to win the game. That is, can their final score be strictly larger than R?

**Input:** There is a single line of input, with three integers, R, O, C.

**Output:** Output in a single line, the answer, which should be "YES" if it's possible for Chef's Team B to win the match and "NO" if not.

- Make Array Zeros using pointers

You are given an array A of length N (size should be created using Dynamic memory allocation) and can perform the following operation on the array:

Select a subarray from array A having the same value of elements and decrease the value of all the elements in that subarray by any positive integer x.

Find the least possible number of operations required to make all the elements of array A equal to zero.

The first line contains an integer N denoting the number of elements in the array.

The next line contains space-separated integers denoting the elements of array A.

Print the least possible number of operations required to make all the elements of array A equal to zero.

Sample Test case

**Input:**

5

2 2 1 3 1

**Output:**

4

## UNIT-2

0L+4T+12P=16 Hours

### PATTERNS:

### PRACTICES:

Problems on Number Patterns

- Write a program to generate Floyd triangle. Sample input N= 4.  
1  
2 3  
4 5 6  
7 8 9 10
- Write a program to generate the following pattern. Sample input N=5.  
13579  
3579  
579  
79  
9
- Write a program to generate the following pattern. Sample input N=4.  
1111111  
222222  
33333  
4444  
333  
22  
1
- Write a program to generate the following pattern. Sample input N=5.  
5432\*  
543\*1  
54\*21  
5\*321  
\*4321
- Write a program to generate the following pattern. Sample input N=5.  
12    21  
123   321  
1234 4321  
123454321
- Write a program to generate the following pattern. Sample input N=5.  
1  
2\*2

3\*3\*3  
 4\*4\*4\*4  
 4\*4\*4\*4  
 3\*3\*3  
 2\*2  
 1

- Write a program to generate the following pattern. Sample input N=4.

1  
 212  
 32123  
 4321234

- Write a program to generate the following pattern. Sample input N=5.

\*  
 \* \*  
 \* \* \* \*  
 \* \*  
 \*

- Write a program to print Pascal triangle for the given number of rows. Sample input N=5.

1  
 1 1  
 1 2 1  
 1 3 3 1  
 1 4 6 4 1

- Write a program to generate the following pattern. Sample input N=4.

1234  
 2341  
 3421  
 4321

- Print Hollow Diamond pattern.
- Print pascals triangle.
- Print Floyds triangle.
- Print Butterfly Pattern.
- Print palindromic pattern.
- Print full inverted number triangle.
- Check if a number is prime or not ( Efficient Approach).
- Find sum of all the digits of the number.
- Print transpose of given matrix.
- Rotate a two dimensional matrix by 90, 180, 270 degrees.

## MODULE-2

### UNIT-1

0L+4T+12P=16 Hours

### ARRAYS

#### PRACTICES:

#### Problems On Arrays

- Given an unsorted array of size N, and the array elements are in the range of 1 to N. There are no duplicates, and the array is not sorted. One of the integers is missing. Write a program to find the missing number.

- Given an array consisting of only 0s and 1s in random order rearrange the array such that all the 0s are to the left of the array and 1s to the right.
- Give an array consisting of odd and even numbers in random order, rearrange the array such that all the odd numbers are to the left of the array and even numbers are to the right of the array.
- Write a program to find all the unique elements in an array.
- Write a program to merge two arrays of the same size sorted in descending order.
- Write a program to count the frequency of each element in an array of integers.
- Write a program to find the second largest element in an array.
- Write a program to find the second smallest element in an array.
- Write a program to find that one element in array that occurs odd number of times, where every other element appears even number of times.
- Create a jagged array (adjacency list representation of a graph) with no of rows and no of columns in each row as specified by the user.

**Hint:** Use Dynamic memory allocation (malloc() or calloc())

**Input:**

Enter no of rows: 3

Enter no of columns Row in 1: 3

Enter no of columns Row in 2: 5

Enter no of columns Row in 3: 2

Enter the elements row wise:

8 6 5

8 4 6 9 7

9 2

**Output:**

8 6 5

8 4 6 9 7

9 2

- Write a program to find second largest number in the array.
- Write a program to find first repeating element in the array.
- Write a program to left rotate the array.
- Write a program to right rotate the array.
- Write a program to find the largest continuous sum.
- Write a program to print the sum of 2nd largest and 2nd smallest elements.
- Write a program to find the maximum product of two numbers multiplies in array (same index should not be used twice).
- Rearrange an array consisting of 1s and 0s such that they are alternatively arranged. Print minimum number of moves required.
- In a given array, find two numbers whose sum equal k.
- Find the difference between positive and negative elements in the array.
- Implement sorting algorithms (Insertion, selection, bubble).

**UNIT-2**

**0L+4T+12P=16 Hours**

**STRINGS:**

**PRACTICES:**

Problems on Strings:

- Write a program to reverse a given string word by word.
- Write a program to find the first occurrence of non-repeating character in the given string.
- Write a program to compress the string as provided in the example.
- Write a program to expand a string as provided in the example.

- Write a program to reverse those words of a string whose length is odd.
- Write a program to check if a given matrix is symmetric or not.
- Write a program to convert all the cases of letter (Lower case -> Upper Case, Upper Case-> Lower Case).
- Write a program to reverse all the words (Not the entire sentence but individual words).
- Find the longest palindrome in a given string.
- Check if two strings are anagrams or not.
- Find minimum number of changes to be done to make a string palindrome.
- Convert Excel sheet name to number ( A-1, B-2, Z-26, AA-27).
- Find number of possible palindromes present in a string.
- Write a C program to read a string s, and determine the number of words in s.

**Example:** s=oneTwoThree

There are 3 words in the string: 'one', 'Two', 'Three'.

- Write a C program that reads a string S and remove all duplicates characters from the given string S.

**NOTE:** 1) Order of characters in output string should be same as given in input string.

2) String S contains only lowercase characters ['a'-'z'].

Example: S = Vignanuniversity

The program should generate the output as: Vignauersty

- Today Ron is reading the book. Due to some reason, he started hating the word 'are' (without quotes). So he decided to replace the substring 'are' with 'R'. Write a C program that reads a line of message 's' and replace the substring 'are' with 'R'. Example: s= Howareyou.

The program should generate the output as: HowRyou

- Write a program to concatenate the characters of the two given strings alternatively.
- Given a string S consisting of uppercase and lowercase letters, change the case of each alphabet in this string. That is, all the uppercase letters should be converted to lowercase and all the lowercase letters should be converted to uppercase.

**Input:** Vignan University

**Output:** vIGNAN uNIVERSITY

- Write a program to insert a given character at the beginning and end of the given string.
- Given two Strings A and B. They are said to be friends if ASCII sum of the each individual string is divisible by 4 else they are not friends. You need to find whether given two strings are friends or not.

Sample Test case:

**Input:**

man nam

vignan university

**Output:**

YES

NO

- Write a program to find the frequency of each digit in the given string.

**Input Format**

The first line contains a string, which is the given number.

**Output Format**

Print ten space-separated integers in a single line denoting the frequency of each digit, indicate that the integers are from 0 to 9.

**Sample Input 0**

a11472o5t6

**Sample Output 0**



0 2 1 0 1 1 1 1 0 0

Explanation 0

In the given string:

1 occurs two times.

2,4,5,6 and 7 occur one time each.

The remaining digits and don't occur at all.

- Sherlock considers a string to be valid if all characters in the given string appear the same number of times. It is also valid if he can remove just 1 character at 1 index in the string, and the remaining characters will occur the same number of times.

Write a C program that reads a string s and determine whether it is valid or not. If valid, return YES, otherwise return NO.

**Example:** S=abc

This is a valid string because frequencies are {a:1,b:1,c:1}

S=abcc

This is a valid string because we can remove one c and have 1 of each character in the remaining string.

S=abccc

This string is not valid as we can only remove 1 occurrence of c. That leaves character frequencies of {a:1,b:1,c:2}

- Read a string containing characters A and B only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string.

Write a C program that finds the minimum number of deletions required.

**Example:** S=AABAAB

Remove A at positions 0 and 3 to make S=ABABA in 2 deletions.

**Input Format**

The first line contains an integer ( the number of queries ).

The next q lines each contain a string s to analyze.

**Sample Input:**

5

AAAA

BBBBB

ABABABAB

BABABA

AAABBB

**Sample Output:**

3

4

0

0

4

- Write a C program that reads a string 's' and it is said to be complete if it contains all the characters from a to z.

**Input Format**

First line of the input contains the number of strings N. It is followed by N lines each contains a single string.

**Output Format**

For each test case print "YES" if the string is complete, else print "NO"

Constraints  $1 \leq N \leq 10$

The length of the string is at max 100 & the string contains only the characters a to z.

- Write a C program that reads two strings and determine whether they share a common substring or not. A substring may be as small as one character.

**Example;**

S1=and

S2=art

The common substring in these two strings: a.

**Sample Input**

2

hello

world

hi

world

**Sample Output**

YES

NO

#### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                  | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Write simple, but complete, C programs.                                                                                                                          | Apply        | 1, 2       | 1                |
| 2      | Identify suitable data type for operands and design of expressions having right precedence.                                                                      | Apply        | 1, 2       | 1                |
| 3      | Apply decision making and iterative features of C Programming language effectively.                                                                              | Apply        | 1, 2       | 1                |
| 4      | Select problem specific data structures and suitable accessing methods.                                                                                          | Analyse      | 1, 2       | 1, 2             |
| 5      | Design and develop non- recursive and recursive functions and their usage to build large modular programs and also able to design string manipulation functions. | Create       | 1, 2       | 3                |
| 6      | Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.                                     | Create       | 1, 2       | 3, 4             |

#### TEXT BOOKS:

1. Behrouz A. Forouzan, Richard F.Gilberg, "Programming for Problem Solving", 1st edition, Cengage publications, 2019.
2. Ajay Mittal, "Programming in C - A Practical Approach", 1st edition, Pearson Education, India, 2010.

#### REFERENCE BOOKS:

1. Reema Thareja, "Computer Fundamentals and Programming in C", 1st edition, Oxford University Press, India, 2013.
2. Herbert Schildt, "C: The Complete Reference", 4th edition, Tata McGraw-Hill, 2017.
3. Byron S Gottfried, "Programming with C", 4th edition, Tata McGraw-Hill, 2018.

# 22EN104 TECHNICAL ENGLISH COMMUNICATION

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 2 | 3 |

**PREREQUISITE KNOWLEDGE:** Basic sentence formation, understanding contextual meanings, basic writing skills and moderate fluency in English.

## COURSE DESCRIPTION AND OBJECTIVES:

In this course students will read, analyze, and interpret material from technical and general fields, and practice reading, writing, listening and speaking skills to gain exposure and functional English on a variety of contemporary topics. The overall course objective is to provide English for Specific Purposes(ESP) instruction to enhance students' reading, writing, listening and speaking skills through a practice in the language. It will aim to build students' confidence and motivation through exposure to academic skills like Note making/taking, Paraphrasing, Summarizing, Report Writing, Making Presentations etc., so as to generate interest in the language from an ESP perspective. Finally, students are expected through the course to gain key strategies and expression for communicating with professionals and non-specialists.

## MODULE-1

### UNIT-1

8L+0T+8P=16 Hours

#### GENETICS:

**Reading:** Reading for Note Making Sub skills: Reading for global understanding (skimming), specific information (scanning), understanding main ideas and supporting ideas, guessing contextual meanings from the text. -Vocabulary building: commonly used roots, prefixes, and suffixes.

**Writing:** Note making, organising main points and sub points, numbering and sequencing, suggesting titles, paraphrasing and summarising.

**Functional grammar:** Common Errors in Articles and Prepositions (Handout).

**Listening:** Listening for Note Taking: top down and bottom up approach, listening for main ideas and supporting points.

**Speaking:** Presentation in teams - ideas on the topic summarised, making a PPT, effective introductions and conclusions, logical organisation of content, using appropriate structure and cohesive devices.

### UNIT-2

8L+0T+8P=16 Hours

#### ALIENS:

**Reading:** Reading: predicting, skimming, scanning, reading for inference, extrapolative reading.

**Vocabulary building:** Academic vocabulary from the text: synonyms, antonyms, Words often confused.

**Writing:** Paragraph writing; writing a topic sentence, supporting sentences, effective introductions and conclusions, use of cohesive devices. Types of Paragraphs: Descriptive, narrative, argumentative and expository.

**Functional grammar:** Common Errors in Verb forms and Conditional sentences (Handout).

**Listening:** Listening for identifying parts from a description, listening to and sorting information, listening for specific information.

**Speaking:** Narrating/Retelling an incident, using suitable cohesive devices/discourse markers Speaking of past and present habits/activities/events - Speaking of future plans.



Source: <https://www.abebooks.com/9781316640081/English-Technical-Communication-Students-Book-1316640086/plp>

**SKILLS:**

- ✓ Apply different sub skills like skimming, scanning, reading for information, reading for inference etc. to understand different kinds of text.
- ✓ Apply different sub skills like top down, bottom up approaches to listening.
- ✓ Use functional vocabulary relevant to engineering and technology to express ideas lucidly.
- ✓ Use appropriate sentence structure, cohesive devices to construct simple text in regular correspondence like e-mails and letters.

**PRACTICES:**

- Note making.
- Summarizing.
- Paragraph Writing.
- Error correction and Restructuring.
- Vocabulary building.
- Listening comprehension.
- Note taking.

**MODULE-2****UNIT-1****8L+0T+8P=16 Hours****SOCIAL MEDIA – HEALTH AND NUTRITION:****Reading:** Reading for factual information researching for supporting evidence - skimming, scanning.**Vocabulary building:** One-word substitutes.**Writing:** Letter Writing- E-mail writing – New age communication – Format, protocol, and style- WhatsApp, Facebook and Twitter Functional grammar: Common Errors in Sub-Verb Agreement and Modals.**Listening:** Listening to a Business Presentation: Listening for deducing information, for abstract details and specific details, listening for taking a message.**Speaking:** Making a presentation with a PPT on a topic assigned- organising the presentation using appropriate discourse markers - presenting a point of view - Extempore.**UNIT-2****8L+0T+8P=16 Hours****FASHION:****Reading:** Reading for data interpretation and information transfer from graphical aids to text reports (pictograms, tables, graphs, pie charts, flow charts), deducing specific information and general information.**Vocabulary building:** business vocabulary, collocations, idioms and phrasal verbs.**Writing:** Writing a Report: Drafting general and factual reports - writing an overview - an effective introduction - organising information into paragraphs (Stages of writing: planning /organising /writing / editing /rewriting).**Functional grammar:** transformations and miscellaneous common errors.**Listening:** Listening to a Ted talk and sorting information - taking notes from a discussion.**Speaking:** Group Discussion - prerequisites -generating content - initiating a discussion - expressing one's opinion - leading a discussion - agreeing/ disagreeing to someone's view - cutting into a speech - body language and voice modulation.**PRACTICES:**

- E-mail writing.
- Letter writing.
- Report writing.
- Messaging in Social media.
- Extempore.
- Making PPTs.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                             | Blooms Level | Module No. | Mapping with POs   |
|--------|-----------------------------------------------------------------------------------------------------------------------------|--------------|------------|--------------------|
| 1      | Apply a variety of strategies to interpret and comprehend spoken texts/ discourse using contextual clues.                   | Apply        | 1          | 6, 7, 8, 9, 10, 12 |
| 2      | Apply appropriate reading strategies to interpret content / material related to engineering and technology domain.          | Apply        | 1          | 6, 7, 8, 9, 10, 12 |
| 3      | Participate in discussions and make short presentations on general and technical topics.                                    | Apply        | 1, 2       | 6, 7, 8, 9, 10, 12 |
| 4      | Possess an ability to write clearly on topics related to technology and workplace communication.                            | Analyze      | 2          | 6, 7, 8, 9, 10, 12 |
| 5      | Choose functional language, grammar structures, cohesive devices and skills of organisation to express clearly in speaking. | Evaluate     | 2          | 6, 7, 8, 9, 10, 12 |

**LANGUAGE LAB ACTIVITIES**

Session - 1: Dictionary Skills.

Session - 2: Introduction to Phonetics and Identifying Phonemes.

Session - 3: Pronunciation Practice (Commonly mispronounced words).

Session - 4: Rosetta Stone (Exercises on LSRW).

Session - 5: Listening Comprehension (Summarising exercise on a Ted Talk).

Session - 6: Technical Presentations (Individual).

Session - 7: Technical Presentations (Team).

Session - 8: TOEFL Mastery.

**TEXT BOOK:**

1. N P Sudharshana & C Savitha, "English For Technical Communication", Cambridge University Press, 2016.

**REFERENCE BOOKS:**

1. Balasubramanian T, "A Text book of Phonetics for Indian Students", Orient Longman, New Delhi, 1989.
2. Krishnaswamy, N and Sriraman, T, "Current English for Colleges", Trinity publications, 2016.
3. Mohan Krishna and Meera Banerjee, "Developing Communication Skills", Macmillan India Ltd. New Delhi, 1990.
4. Ashraf Rizvi M, "Effective Technical Communication", 2nd Edition, McGraw Hill Education, 2017.
5. Narayana Swamy V R, "Strengthen your Writing", Third Edition Orient Black Swan, New Delhi, 2005.

# 22CS104 PYTHON PROGRAMMING

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 2 | 3 |

**PREREQUISITE KNOWLEDGE:** Prior knowledge of any programming language and object-oriented concepts is helpful but not mandatory.

## COURSE DESCRIPTION AND OBJECTIVES:

This course offers sufficient knowledge required to understand the fundamental concepts of Python programming language. This course enables students to choose appropriate data structures (lists, dictionaries, tuples, sets, strings) for the given problem. In addition, the students will be able to create reliable, modular and reusable applications using Object- Oriented Programming approaches. At the end they will get an idea of how to access database using python programming, develop web applications, and using web Services using python Programming.

## MODULE-1

### UNIT-1

6L+0T+6P=12 Hours

#### INTRODUCTION:

Introduction to python, Variables, Assignment, Keywords, Built-in functions, Indentation, Comments, Basic data types - integers, float, complex, Boolean, strings; Python program development, running python using REPL shell, Python scripts.

**Operators and Expressions:** Operators- arithmetic operators, comparison (relational) operators, assignment operators, logical operators, bitwise operators, membership operators, identity operators; Expressions and order of evaluations.

**Control Structures:** Conditional control structures - if, elif, else; Loop control structures - for, while, for... else, while...else, nested loops, break, continue, pass.

### UNIT-2

10L+0T+10P=20 Hours

#### PYTHON DATA STRUCTURES AND FUNCTIONS:

**Data Structures:** Lists, Tuples, Sets, Strings, Dictionary - creation, accessing, basic operators and methods; List comprehensions.

**Functions:** Defining functions, calling functions, Passing arguments - keyword arguments, default arguments, positional arguments, variable-length arguments; Types of functions- anonymous functions, fruitful function, recursive functions; Scope of the variables- global and local variables, Development of recursive and non-recursive functions.

#### PRACTICES:

- A. Given an integer N, write a program to find its 1's complement.
- B. Given two integers N1 and N2, write a program to find their product without using multiplication (\*) operator and loops.
- C. Given two integers N1 and N2 having same value, write a program to check whether N1 and N2 points to the same object or not.
- A. Given an Integer N, write a program to check whether given number is even or odd without using modulus operator.
- Given a number N, number of bits K and starting position P, write a program to extract K bits from a position P (from right) in the binary representation of N. Convert the extracted bits in decimal number.

Source: <https://www.hitalent.co/blog/2019/12/tech-jobs-python-programming-language-and-aws-skills-demand-has-exploded>

- Given coordinates of centre of a circle, radius and a point coordinate, write a program to check whether the given point lies inside or on the circle, or outside the circle.
- Write a program to find the sum of digits in a given integer.  
Given an integer N as an input, decides the geometrical figure for which the area has to be calculated, for example N=1 for circle, N=2 for rectangle, and N=3 for triangle. Write a program to display the area of the respective figure.
- A semi prime number is an integer which can be expressed as a product of two distinct primes. For example,  $15 = 3 \times 5$  is a semi prime number but  $9 = 3 \times 3$  is not.  
For a Given an integer number N, write a program to find whether it can be expressed as a sum of two semi-primes or not (not necessarily distinct).
- Given an integer amount X, write a program to find the minimum number of currency notes \$ (500, 100, 50, 20, 10, 5, 2, 1) required for the given amount.

**Input:**

575

Where input is the amount for which we have to calculate the number of currency notes.

**Output: 4****Explanation:** Total amount = 1(500 dollar note) + 1(50 dollar note) + 1(20 dollar note) + 1(5 dollar note) = 575, hence the minimum number of notes required is 4.

- For a given a string S and width W, write a program to wrap the string S into a paragraph of width W.

**Example:****Input:**

ABCDEFGHIJKLMNOQRSTUVWXYZ

4

**Output:**ABCD EFGH IJKL IMNO QRST UVWX  $\Sigma$ 

- Write a program to Measure the required time to access the first element, nth element and n/2 element stored in list and tuple data structure.  
Given a list L of N numbers (integers), Write a program to find the sum of the elements of given list L with the corresponding elements of the reverse of list L. If list L has elements [1,2,3], then reverse of the list L will be [3,2,1] and the resultant list should be [4,4,4].
- Given a positive integer number n. Write a program to generates a dictionary d which contains (i, i\*i\*i) such that i is the key and i\*i\*i is its value, where i is from 1 to n (both included). Print the content of the dictionary d.
- Write a program to create a data structure to store student information such as regd no, name, percentage of marks, phone number and display the student details based on the order of percentage of marks.
- For a Given a string, design and implement functions to perform the following:
  - remove vowels in the given string.
  - count number of uppercase and lowercase letters.
  - remove all special characters.
  - check whether it is a palindrome or not.
  - swap case of each letter.
- Create a function that receives 3 numbers and returns the median, i.e. the number that is not the min and not the max, but the one in between.
- Given two lists of integer numbers, write a function to perform the following operations.
  - print elements that are common in both the lists. (Print without duplicates).
  - print elements that are present in the first list and not present in second list.
  - print elements that contain the first element of the first list and last element of the second list.
  - print elements that contain sum of elements of first list and sum of elements of second list.

**SKILLS:**

- ✓ *Installation and usage of python libraries.*
- ✓ *Working with varieties of data structures.*
- ✓ *Improved analytical and problem-solving abilities.*
- ✓ *Developing structured modular and Object-oriented programming solutions.*

- e. print largest number of both the lists.
- f. print smallest number of both the lists.

## MODULE-2

### UNIT-1

**8L+0T+8P=16 Hours**

#### MODULES:

Creating modules, Import Statement, From...Import Statement, Name Spacing, Creating user defined modules.

**Standard Modules:** sys, math, date, time, os, random and itertools modules.

**Packages:** Numpy, Pandas, Matplotlib, Requests, Nltk.

**File Processing:** Reading and writing files -creating a new file, writing to a file, reading text files, opening and closing files, reading, writing, tell (), seek (), rename ().

### UNIT-2

**8L+0T+8P=16 Hours**

#### ERRORS AND EXCEPTIONS:

Introduction to Exceptions, Handling Exception, Try Except Else and Finally Block, Raising Exceptions.

**Simple Graphics and Image Processing:** Overview of Turtle Graphics, Two Dimensional Shapes, Colours and RGB System and Image Processing

#### PRACTICES:

- Given a string 'S', find all possible permutations of the string S in lexicographic sorted order. Each Permutation size is "2" or "3".

**Sample Input:**

HACK

**Expected Output:** AC AH AK CA CH CK HA HC HK KA KC KH

- Write a program that finds area of the pentagon when length from center of a pentagon to vertex are given, the formula for computing the area of pentagon is  $\frac{3}{2} s^2$ , where s is the length of the side, the side can be computed using formula  $s = 2r \sin\left[\frac{\pi}{5}\right]$ , where r is the length from the center of a pentagon to vertex.

Given X as a date. Write a program to find what the day is on that date.

**Sample Input:** 08 05 2015

**Expected Output:** Wednesday

- Arun is working in an office which is N blocks away from his house. He wants to minimize the time it takes him to go from his house to the office. He can either take the office cab or he can walk to the office. Arun's velocity is V1 m/s when he is walking. The cab moves with velocity V2 m/s but whenever he calls for the cab, it always starts from the office, covers N blocks, collects Arun and goes back to the office. The cab crosses a total distance of N meters when going from office to Arun's house and vice versa, whereas Arun covers a distance of  $(2-\sqrt{2})(2*N)$  while walking. Help Arun to find whether he should walk or take a cab to minimize the time.

**Input Format:**

A single line containing three integer numbers N, V1, and V2 separated by a space.

**Example-1:**

**Input:**

5 10 15

**Output:**

Cab

- Create a binary NumPy array (containing only 0s and 1s) and convert a binary NumPy array into a Boolean NumPy array Convert the first column of a Data Frame as a Series by using suitable packages.



**Sample Input:**

Original Data Frame

```
col1 col2 col3
0 1 4 7
1 2 5 5
2 3 6 8
3 4 9 12
4 7 5 1
5 11 0 11
```

**Sample Output:**

```
0 1
1 2
2 3
3 4
4 7
11
```

- Create two text files and read data from two text files. Display a line from first file followed by the corresponding line from the second file.
- Define the following functions that are more robust to erroneous input data
  - a) To divide two numbers (To handle Zero Division Error).
  - b) To read two integer numbers and display them (To handle Value Error).
  - c) To display elements of a list (To handle Index Error).
  - d) To open a file and display file contents (To handle File Not Found Error).
- Write a python program to handle multiple exceptions using raise keyword.
- Draw the spiral hexagon, where we use turtle to create a spiral structure. The final shape is a hexagon and there are various colors used in producing the sides of the hexagon.
- Implement a program to print it in a counterclockwise spiral form for a given square matrix.

**Sample Input:**

```
4
25 1 29 7
24 20 4 32
16 38 29 1
48 25 21 19
```

**Sample Output:**

```
25 24 16 48 25 21 19 1 32 7 29 1 20 38 29 4
```

- Write a function that finds the nearest prime number of a given number.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                             | Blooms Level | Module No. | Mapping with POs |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Analyze various features of programming language and their application in problem solving in computer programming to write, compile, and debug programs in python language. | Analyze      | 1          | 1, 2             |
| 2      | Make use of different data types to design programs involving decisions, loops, and functions.                                                                              | Apply        | 1          | 1, 2, 5          |
| 3      | Analyze the usage of different data structures for practical and contemporary applications for a given problem.                                                             | Analyze      | 1          | 1, 2, 3, 5       |
| 4      | Develop functional, reliable and User-friendly Python programs for given problem statement and constraints.                                                                 | Apply        | 2          | 1, 2, 3, 5       |
| 5      | Installing the python environment and related packages that are required for practical and contemporary applications.                                                       | Apply        | 2          | 1, 2, 3, 5       |

**TEXT BOOKS:**

1. Kenneth A. Lambert, "The Fundamentals of Python: First Programs", Cengage Learning, 2011.
2. Mark Lutz, "Learning Python", 5th edition, Orielly Publishers, 2013.

**REFERENCE BOOKS:**

1. Introduction to Computation and Programming Using Python. John V. Guttag, The MIT Press.
2. James Payne, "Beginning Python using Python 2.6 and Python 3", Wrox publishing.
3. Paul Gries, "Practical Programming: An Introduction to Computer Science using Python 3", The Pragmatic Bookshelf, 2nd edition, 4 Oct. 2013.
4. Allen B. Downey, "Think Python", 1st edition, Orielly publishing.

**II**  
YEAR

**B.Tech.**

**COURSE CONTENTS**

I SEM & II SEM

# CSE - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

## I SEMESTER

|   |         |                                                                      |
|---|---------|----------------------------------------------------------------------|
| ▶ | 22ST203 | - Probability Theory and Statistics for Machine Learning             |
| ▶ | 22MS201 | - Management Science                                                 |
| ▶ | 22TP202 | - Data Structures using Python                                       |
| ▶ | 22AM201 | - Artificial Intelligence                                            |
| ▶ | 22AM202 | - Digital Logic and Computer Organization                            |
| ▶ | 22CS201 | - Database Management Systems                                        |
| ▶ | 22CT201 | - Environmental Studies                                              |
| ▶ | 22SA201 | - Life Skills - I                                                    |
| ▶ |         | - NCC/NSS/SAC/E-Cell/Student Mentoring/Social activities/Publication |

## II SEMESTER

|   |         |                                                  |
|---|---------|--------------------------------------------------|
| ▶ | 22TP203 | - Advanced Coding Competency                     |
| ▶ | 22TP204 | - Professional Communication                     |
| ▶ | 22AM203 | - Design and Analysis of Algorithms using Python |
| ▶ | 22AM204 | - Machine Learning                               |
| ▶ | 22CS203 | - Object-Oriented Programming through JAVA       |
| ▶ | 22CS207 | - Operating Systems                              |
| ▶ | 22SA202 | - Life Skills - II                               |



# 22ST203 PROBABILITY THEORY AND STATISTICS FOR MACHINE LEARNING

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |



Source: <https://www.amazon.com/Differential-Equations/dp/B01H30X2JA>

**PREREQUISITE KNOWLEDGE:** Basic mathematical calculations.

## COURSE DESCRIPTION AND OBJECTIVES:

This course is mainly deals with the essential components of Machine learning i.e probability theory and statistics. It provides basic knowledge about probability, random variables, discrete and continuous distributions, sampling distributions. It also deals with sampling techniques for data collection, and introduction to statistical inference methods for decision making and hypothesis testing. The overview of hypothesis for small and large samples, curvefitting, regression and correlation also provided in this course.

## MODULE-1

### UNIT-1

**12L+0T+8P=20 Hours**

#### INTRODUCTION TO MACHINE LEARNING:

**Introduction:** Human learning and its types, machine learning and its types, applications of machine learning, machine learning activities, data types in machine learning, structure of data, central tendency, data spread, measuring data value position.

**Basic Probability:** Random experiments, the concept of probability, axioms of probability, conditional probability, theorems on probability, conditional probability, independent events, Bayes' theorem, combinatorial analysis.

**Random variables:** Random variables, types of random variables, jointly distributed random variables, independent random variables, conditional distributions.

### UNIT-2

**12L+0T+8P=20 Hours**

#### MATHEMATICAL EXPECTATIONS AND DISTRIBUTIONS:

**Expectation:** Expectation, expected value and variance of sums of random variables, standard deviation, covariance and variance of sums of random variables, moment generating functions, Chebyshev's inequality and the weak law of large numbers, other measures of central tendency, percentiles, measures of dispersion, skewness and kurtosis.

**Probability Distributions:** The binomial distribution and properties, the law of large numbers for Bernoulli trials, normal distribution and properties, Poisson distribution and properties, relationships among binomial, normal and Poisson distributions, central limit theorem. multinomial, hypergeometric, uniform, Cauchy, gamma, beta.

#### PRACTICES:

- Describe the data using different statistical measures and graphical display of data.
- Compute and plot the frequency polygon for a given data.
- Measuring the chance of uncertainty of random variables.
- Compute mean and variances for discrete and continuous distributions.
- Find the MGF for various distributions.

**SKILLS:**

- ✓ Identify probability models based on the theoretical results presented in the course.
- ✓ Analyse the data using measures of central tendency.
- ✓ Test the statistical data for rejection or acceptance.

**MODULE-2****UNIT-1****12L+0T+8P=20 Hours****SAMPLING THEORY:**

Population, sample, statistical inference, random samples, sample statistics sampling distributions, sample mean, sampling distributions - means, proportions, differences and sums, sample variance, sampling distribution of variances, sampling distribution of ratios of variances, chi-square, student's t, and F distributions, relationships among chi-square, t, and f distributions. Frequency distributions, relative frequency distributions computation of mean, variance, and moments for grouped data.

**UNIT-2****12L+0T+8P = 20 Hours****CURVE FITTING, REGRESSION, AND CORRELATION:**

Curvefitting, regression, method of least squares, least-squares line, the least-squares parabola, multiple regression, standard error of estimate, linear correlation coefficient, generalized correlation coefficient, rank correlation, probability interpretation of regression and correlation, sampling theory of regression and correlation, correlation and dependence.

**PRACTICES:**

- Fit the various probability distributions for the appropriate data.
- Conducting Correlation and regression analysis for a given data.
- Analyse the sample data to accept or reject statements regarding population parameters.
- Fit an appropriate curve for a given set of data.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                            | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Familiarize with foundations of Human learning and Machine learning and apply Machine Learning to basic types of data.     | Apply        | 1          | 1, 2             |
| 2      | Apply probability and statistical methods to discrete and continuous random variables.                                     | Apply        | 1          | 1, 2, 3          |
| 3      | Apply the concepts in probability distributions and mathematical expectation in machine learning applications.             | Apply        | 1          | 1, 2             |
| 4      | Execute concept of correlation analysis and least square method in fitting regression curves to machine learning problems. | Analyze      | 2          | 1, 2, 3, 4       |
| 5      | Perform Hypothesis test for samples from finite and infinite populations.                                                  | Evaluate     | 2          | 1, 2, 4          |

**TEXT BOOKS:**

1. I.R. Miller, J.E. Freund and R. Johnson, "Probability and Statistics for Engineers", 9th Edition, Pearson, 2018.
2. John Schiller, R. Alu Srinivasan, Murray Spiegel, "Schaum's Outline of Probability and Statistics" 3rd Edition, The McGraw-Hill Companies Inc, 2009.

**REFERENCE BOOKS:**

1. Anirban Das Gupta, "Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics", Springer, 2011.
2. Himanshu Singh, Statistics for Machine Learning, BPB publications, 2021.
3. S. C. Gupta and V. K. Kapoor, "Fundamentals of Applied Statistics", Sultan Chand & Sons, 4th edition, 2007.

**22MS201 MANAGEMENT SCIENCE**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 0 | 3 |

**PREREQUISITE KNOWLEDGE:** Basic knowledge on management .**COURSE DESCRIPTION AND OBJECTIVES:**

The goal of this course is to analyse the importance of management, significance of operation management and carry out production operations through work-study. Students will be able to analyse the markets, customers, competitors, and then plan HR function effectively.

**MODULE-1****UNIT-1****6L+6T+0P=12 Hours****INTRODUCTION TO MANAGEMENT:**

Concepts of Management and organization- nature, importance and Functions of Management, Systems approach to Management - Taylor's Scientific Management Theory, Fayol's Principles of Management, Mayo's Hawthorne Experiments, Maslow's Theory of Human Needs, Douglas McGregor's Theory X and Theory Y, Herzberg's Two-Factor Theory of Motivation, Leadership Styles, Social responsibilities of Management.

**UNIT-2****10L+10T+0P=20 Hours****OPERATIONS MANAGEMENT:**

Principles and Types of Plant Layout-Methods of production (Job, batch and Mass Production), Work Study -Basic procedure involved in Method Study and Work Measurement, Objectives, Need for Inventory control, EOQ, ABC Analysis, Purchase Procedure, Stores Management and Stores Records. Statistical Quality Control: control charts for variables and attributes (simple Problems), Acceptance Sampling.

**PRACTICES:**

- Collect some examples with videos for types of production.
- Carry out production operations through work-study.
- Practice problems with Inventory control methods and Quality Control charts.

**MODULE-2****UNIT-1****8L+8T+0P=16 Hours****HUMAN RESOURCES MANAGEMENT:**

Concepts of Human Resource Management, Basic functions of HR Manager: Manpower planning, Recruitment, Selection, Training and Development, Placement, Wage and Salary Administration, Promotion, Transfer, Separation, Performance Appraisal, Grievance Handling and Welfare Administration, Job Evaluation and Merit Rating.

**UNIT-2****8L+8T+0P=16 Hours****MARKETING MANAGEMENT:**

Evolution of Marketing, Functions of Marketing Selling Vs Marketing, 4 P's of Marketing - Product Mix - Product Life Cycle - Place Mix - Channels of Distribution - Price Mix - Pricing Methods - Promotion Mix - Tools of Promotions.



Source: <https://xueqi326.wordpress.com/semester-3/management-science/>

**SKILLS:**

- ✓ *Expert in managerial skills.*
- ✓ *Maintain social relations.*
- ✓ *Evaluate pricing strategies.*

**PRACTICES:**

- Select any Designation in an organization and try to describe its job description and job specifications.
- How do you deal with grievances at your work.
- Analyze marketing mix in various situations.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                    | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------|--------------|------------|------------------|
| 1      | Analyze the nature and importance of management    | Analyze      | 1          | 1, 2, 4, 6       |
| 2      | Significance of Operations Management.             | Analyze      | 1, 2       | 1, 2, 5          |
| 3      | Carry out production operations through work-study | Apply        | 1, 2       | 1, 2, 3, 5       |
| 4      | Analyze the markets, customers, and competition    | Analyze      | 2          | 1, 2, 4, 5, 6    |
| 5      | Plan and control the HR function effectively       | Evaluate     | 1, 2       | 1, 2, 3, 4, 5, 6 |

**TEXT BOOKS:**

1. Stoner, Freeman, Gilbert, Management, 6th Ed, Pearson Education, New Delhi, 2004.
2. P. Vijay Kumar, N. Appa Rao and Ashnab, Chnalill, Cengage Learning India, 2012.

**REFERENCES :**

1. Kotler Philip & Keller Kevin Lane: Marketing Mangement 12/e, PHI, 2005.
2. Koontz & Weihrich: Essentials of Management, 6/e, TMH, 2005.
3. Thomas N. Duening & John M .Ivancevich Management - Principles and Guidelines, Biztantra, 2003.
4. Aryasri: Management Science, TMH, 2004.



# 22TP202 DATA STRUCTURES USING PYTHON

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Programming in C.

## COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed at offering fundamentals concepts of data structures and explains how to implement them. It begins with the basic concepts of data, data structures and then introduces the primitive and non-primitive data structures in detail. It forms the basis for understanding various ways of representing data and its usage in different computing applications.

## MODULE-1

### UNIT-1

6L+6T+6P=18 Hours

#### DATA STRUCTURES BASICS:

**Basic Terminology** – data, information, datatype; Data Structures – Introduction, storage structures-sequential and linked storage representations; classification of data structures; Applications of data structures.

**Sorting:** Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort.

**Searching:** Linear Search and Binary Search.

### UNIT-2

10L+10T+10P=30 Hours

#### LINKED LISTS AND STACKS, QUEUES:

**Linked List:** Introduction, Types of linked list – Singly linked list, doubly linked list and circular linked list, representation of linked list, Operations of linked list: Traverse forward/ reverse order, searching, insertion and deletion; Applications of linked lists.

**Stack** – Introduction, array and linked representations, implementation and their applications; Queue – Introduction, array and linked representations, implementation; Types – Linear, circular and doubly ended queues – operations; Applications of Queues.

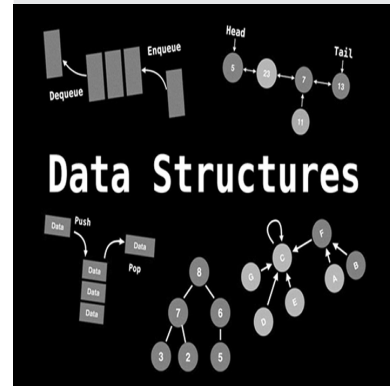
#### PRACTICES:

##### PROBLEMS ON RECURSION – LEVEL 1

- Find the product of 2 numbers using recursion.
- Find the sum of natural numbers using recursion.
- Find the factorial of a number using recursion.
- Find the Nth term of Fibonacci series using recursion.
- Calculate the power using recursion.
- Write a recursive program for checking if a given number is a prime number.
- Given two integers write a function to sum the numbers without using any arithmetic operators.
- Convert a decimal to binary using recursion.
- Print all factors using recursion.
- Find the maximum product of digits among numbers less than or equal to N.

##### PROBLEMS RECURSION – LEVEL 2

- Implement insertion sort recursively.



Source:  
<https://www.youtube.com/watch?v=Qmt0QwzEmh0>

**SKILLS:**

- ✓ Experienced to Store data and various types of data to handle.
- ✓ Ordering and sorting of data.
- ✓ Indexing and Searching of required data from large data sequences.
- ✓ Exposed to various characteristics such as Linear or non-linear, Homogeneous or heterogeneous and Static and Dynamic.

- Write a program to find the numbers less than N that are product of exactly 2 distinct prime numbers - using recursion.
- Implement selection sort recursively.
- Find the middle of a singly linked list using recursion.
- Find the sum of even numbers of an array using recursion.
- Check if a given array is in sorted order using recursion.
- Print alternate nodes of a linked list using recursion.
- Reverse a doubly linked list using recursion.
- Write a recursive function that returns all permutations of a given list.
- Implement bubble sort recursively.

**PROBLEMS ON SORTING AND SEARCHING – LEVEL 1**

- Implement the insertion sort function.
- Implement the bubble sort function.
- Implement the quick sort function.
- Implement the merge sort function.
- Implement the selection sort function.
- Implement linear search function.
- Implement binary search function.

**PROBLEMS ON SLL – LEVEL 1**

- Implement the insert function to insert nodes into a singly linked list (ascending order).
- Implement the insert function to insert nodes into a singly linked list (descending order).
- Implement the search node function.
- Implement the delete node function.
- Display forwards function.
- Display backwards function.
- Count the number of nodes in a singly linked list.
- Swap alternate nodes of a singly linked list.
- Move last node to the front of the linked list.
- Move first node to the last of the linked list.

**PROBLEMS ON STACKS – LEVEL 1**

- Implement two stacks using a single array.
- Given an array replace every element with nearest greater element on the right.
- Given a stack reverse the elements using only push and pop functions.
- Postfix evaluation using stack.
- Balance symbols.
- Find middle element in a stack.
- Remove middle element from a stack.
- Implement push and pop using linked list.
- Given an array of characters with the middle marked by X, check if the string is a palindrome.
- Maximum sum in sliding window.

**PROBLEMS ON QUEUES – LEVEL 1**

- Write a program to accept two numbers as input check if they are equal.
- Write a program to accept two characters as input and check if they are equal.
- Write a program to accept two numbers as input and print the greater of the 2 numbers.
- Write a program to accept two numbers as input and print the lesser of the 2 numbers.
- Write a program to accept 3 numbers as input and print the maximum of the 3.
- Write a program to accept 3 numbers as input and print the minimum of the 3.
- Write a program to accept a number as input and print EVEN if it is an even number and ODD if it is an odd number.

- Write a program to accept a number as input and check if it is divisible by 3. If it is divisible by 3 print YES else print NO.
- Write a program to accept a number as input and check if it is divisible by both 3 & 5. If it is divisible print YES else print NO.
- Write a program to accept a number as input and check if it is positive, negative or zero.

**PROBLEMS ON DLL – LEVEL 1**

- Implement insert function.
- Implement display forward function.
- Implement display backward function.
- Implement search function.
- Implement delete function.
- Reverse a doubly linked list from M to N.
- Find the sum of the odd and even nodes.
- Count odd keys of the linked list.
- Merge two sorted lists.
- Delete adjacent duplicate nodes.

**PROBLEMS ON CLL – LEVEL 1**

- Insert function (circular doubly linked list).
- Search function.
- Display forward.
- Display backward.
- Delete node (circular doubly linked list).
- Print the middle N nodes of a circular singly linked list.
- Move the last node of a circular singly linked list to the beginning.
- Delete adjacent duplicate nodes of a circular singly linked list.
- Delete nodes greater than a value from a circular doubly linked list.
- Find the sum of the nodes of a circular linked list.

**PROBLEMS ON LINKED LIST – LEVEL 2**

- Given 2 sorted linked lists, print the common elements.
- Reverse a list (using Stack).
- Given a pointer to a node (not the last node), delete the node.
- Reverse a list (Recursive).
- Reverse a list (Iterative).
- Reverse a singly linked list in pairs (recursive).
- Reverse a singly linked list in pairs (iterative).
- Check if a singly linked list is a palindrome or not.
- Remove the loop if exists.
- Given 2 linked lists with data in the ascending order, merge them into a single list.

**MODULE-2****UNIT-1****8L+8T+8P=24 Hours****TREES:**

**Trees:** Basic Terminology, Types of Trees, Binary Tree – Introduction, properties, array and linked representations; Tree traversals and their implementation; Expression trees; BST – definition and operations, AVL trees – definition and construction; Applications of binary trees.

**UNIT-2****8L+8T+8P=24 Hours****GRAPHS & HASHING:**

**Graphs:** Basic Terminology, Types of Graphs, Graphs representations – adjacency matrix, adjacency

list; Traversals - breath first search and depth first search; Applications of graphs.

**Hashing:** Introduction, Different hash functions, collision: avoidance and handling methods.

### PRACTICES:

#### PROBLEMS ON BST – LEVEL 1

- Insert function.
- Insert function (recursive).
- Search function.
- Pre order traversal.
- Post order traversal.
- In order traversal.
- Level order traversal.
- Delete child node.
- Delete parent node.
- Delete nodes greater than a value from a circular doubly linked list.

#### PROBLEMS ON PRIORITY QUEUES – LEVEL 1

- Meeting rooms problem.
- Ugly number.
- Find median from data stream.
- Find the top K frequent elements.
- Find K Pairs with smallest sums.
- Find the Kth smallest element in a sorted matrix.
- Trapping Rain Water.
- Rearrange String k distance apart.
- Sort characters by frequency.
- Solve the maze problem.

#### PROBLEMS ON GRAPHS – LEVEL 1

- Implement Graph data structure.
- Implement BFS - iterative solution.
- Implement BFS - recursive solution.
- Implement DFS - iterative solution.
- Implement DFS - recursive solution.
- Check if given graph is strongly connected or not.
- Check if given graph is strongly connected or not - using DFS.
- Given a graph find the arrival and departure time of its vertices in DFS. Arrival time is the time when the vertex was explored for the first time, and departure time is the time at which all the neighbours are explored and are ready to backtrack.
- Given a directed acyclic graph and a source vertex, find the cost of the shortest path from source vertex to all other vertices present in the graph. If a vertex cannot be reached from given source vertex that distance may be printed as infinite.
- Given an undirected graph, check if the graph is 2 edge connected or not.

#### PROBLEMS ON HASHING – LEVEL 1

- Print a binary tree in vertical order.
- Find whether an array is subset of another array.
- Given an array A [ ] and a number x, check for pair in A [ ] with sum as x.
- Minimum operation to make all elements equal in array.
- Maximum distance between two occurrences of same element in array.
- Check if a given array contains duplicate elements within k distance from each other.
- Find duplicates in a given array when elements are not limited to a range.

- Most frequent element in an array.
- Smallest subarray with all occurrences of a most frequent element.
- First element occurring k times in an array.

### PROBLEMS ON GRAPHS – LEVEL 2

- Find the shortest graph distances between every pair vertex in a given path. Assume that the graph does not have any negative edges.
- Find the shortest graph distances between every pair of vertices in a given path. The graph can have negative edges.
- Detect cycle in DFS.
- Count the number of connected components of a graph represented in the adjacent matrix.
- Count the number of connected components of a graph represented in the adjacent matrix - using DFS.
- Find a spanning tree - not necessarily a minimum spanning tree.
- Detect cycle in an undirected graph.
- Given an undirected graph, find its depth.
- Determine if a directed graph has a unique topological ordering.
- Given a directed acyclic graph and two vertices v and w, find the lowest common ancestor.

### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                       | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Explore the organization of several ADTs and the manipulation (searching, insertion, deletion, traversing) of data stored in various data structures. | Apply        | 1, 2       | 1                |
| 2      | Apply different data structures to solve a given problem.                                                                                             | Apply        | 1, 2       | 1                |
| 3      | Analyze the efficiency of using different data structures and choose the efficient data structure for solving a given problem.                        | Analyze      | 1, 2       | 2                |
| 4      | Develop new algorithms to solve various problems.                                                                                                     | Create       | 1, 2       | 3, 4             |

### TEXT BOOKS:

1. Reema Thareja, "Data Structures Using C", 2nd Edition, Oxford University Press, 2014.
2. Seymour Lipschutz, "Data Structures with C", 1st Edition, McGraw Hill Education, 2017.

### REFERENCE BOOKS:

1. Ellis Horowitz and Sartaj Sahni, "Fundamentals of Data Structures", illustrated edition, Computer Science Press, 2006.
2. Richard F. Gilbert and Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", 2nd Edition, CENAGE Learning, 2005.
3. R G Dromey and Pearson, "How to solve it by Computer", 2nd edition, Impression edition, 1998.

# 22AM201 ARTIFICIAL INTELLIGENCE

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Probability & statistics.

## COURSE DESCRIPTION AND OBJECTIVES:

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. In addition to this, student will understand the building blocks of AI such as search, knowledge representation, inference, logic and learning. This course enables the students to develop a small AI system for real time problems.

## MODULE-1

### UNIT-1

**6L+6T+6P=18 Hours**

#### INTELLIGENT SYSTEMS:

Introduction, what is AI, Examples of AI systems, Brief history of AI Agent, Agents and environments, Structure of agents, the concept of rationality, the nature of environments, Types of agents, problem solving approaches to typical AI problem.

### UNIT-2

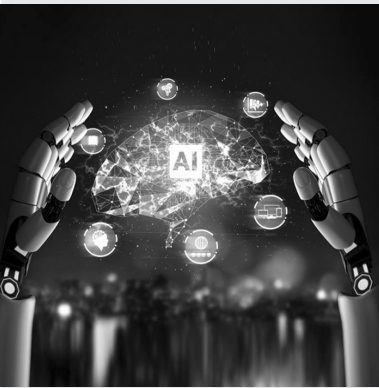
**10L+10T+10P=30 Hours**

#### PROBLEM SOLVING:

State Space Problem; Searching: Uniform search, InformedSearch: Solving problems by searching: Heuristic functions, Hill climbing, Best First Search, A\* algorithm, AO\* algorithm, Searching game trees: Min Max Search, Alpha Beta pruning.

#### PRACTICES:

- In the classical vacuum cleaner problem, we have two rooms and one vacuum cleaner. There is dirt in both the rooms and it is to be cleaned. The vacuum cleaner is present in any one of these rooms. Find the solution, how we can reach to reach a state in which both the rooms are clean and are dust free.
- In this problem, three missionaries and three cannibals must cross a river using a boat which can carry at most two people, under the constraint that, for both banks, that the missionaries present on the bank cannot be outnumbered by cannibals. The boat cannot cross the river by itself with no people on board. Find the solution, how to solve the given problem.
- You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. Find the solution, how can you get exactly 2 gallons of water in the 4-gallon jug?
- There is a farmer who wishes to cross a river but he is not alone. He also has a goat, a wolf, and a cabbage along with him. There is only one boat available which can support the farmer and either of the goat, wolf or the cabbage. So at a time, the boat can have only two objects (farmer and one other). But the problem is, if the goat and wolf are left alone (either in the boat or onshore), the wolf will eat the goat. Similarly, if the Goat and cabbage are left alone, then goat will eat the cabbage. The farmer wants to cross the river with all three of his belongings: goat, wolf, and cabbage. What strategy he should use to do so?
- Either place a block that doesn't have other blocks stacked on top of it on another block with the same behavior, or on the table. The initial and the goal state are described by the exact position of each block. Find the solution, how to solve the given problem.



Source: <https://www.forbes.com/sites/bernardmarr/2020/08/03/3-important-ways-artificial-intelligence-will-transform-your-business-and-turbo-charge-success/>

- Given a 3x3 board with 8 tiles (every tile has one number from 1 to 8) and one empty space. The objective is to place the numbers on tiles to match the final configuration using the empty space. We can slide four adjacent (left, right, above, and below) tiles into the empty space. Find the solution, how to solve the given problem by using A\* search algorithm.
- The rules of tic-tac-toe on the 3x3 field are as follows. Before the first turn all the field cells are empty. The two players take turns placing their signs into empty cells (the first player places Xs, the second player places Os). The player who places Xs goes first, another one goes second. Find the solution, how to solve the given problem where the winner is the player who first gets three of his signs in a row next to each other (horizontal, vertical or diagonal).
- In crypt arithmetic problem, the digits (0-9) get substituted by some possible alphabets or symbols. The task in crypt arithmetic problem is to substitute each digit with an alphabet to get the result arithmetically correct. Find the solution, how to solve the given problem, where we can perform all the arithmetic operations on a given crypt arithmetic problem.

**SKILLS:**

- ✓ Analyze Intelligent systems.
- ✓ Apply problem solving techniques.
- ✓ Interface various knowledge representation.
- ✓ Create a dynamic planning.

**MODULE-2**

**UNIT-1**

**10L+10T+10P=30 Hours**

**KNOWLEDGE REPRESENTATION&PLANNING:**

**Propositional logic:** Inference in propositional logic, Resolution, Forward chaining, Backward chaining, First order logic: Reasoning patterns in First order logic, Resolution, Forward chaining, Backward chaining, The planning problem: Planning with state space search, Partial order planning, Planning graphs.

**UNIT-2**

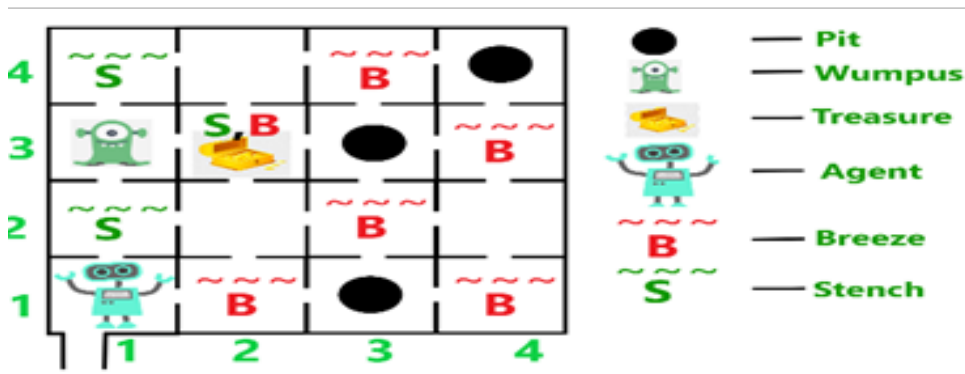
**6L+6T+6P=18 Hours**

**LEARNING:**

**Forms of learning:** Supervised Learning, Unsupervised learning, Reinforcement learning, Learning Decision Trees, Ensemble Learning, Expert system.

**PRACTICES:**

- With logic programming, compare expressions and find out unknown values.
- The Wumpus world is a cave with 16 rooms (4x4). Each room is connected to others through walkways (no rooms are connected diagonally). The knowledge-based agent starts from Room [1, 1]. The cave has – some pits, a treasure and a beast named Wumpus. The Wumpus cannot move but eats the one who enters its room. If the agent enters the pit, it gets stuck there. The goal of the agent is to take the treasure and come out of the cave. The agent is rewarded, when the goal conditions are met. The agent is penalized, when it falls into a pit or being eaten by the Wumpus. Some elements support the agent to explore the cave, like -The Wumpus’s adjacent rooms are stench. -The agent is given one arrow which it can use to kill the Wumpus when facing it (Wumpus screams when it is killed). – The adjacent rooms of the room with pits are filled with breeze. -The treasure room is always glittery. Find the Wumpus presented room.



- You are on one side of a river with a wolf, a goat, and a cabbage. You want to transport all three to the other side of the river, but you can only transport one object at a time. You cannot leave the wolf and the goat alone, or the cabbage and the goat alone; you are the only thing keeping them from eating each other. How can you transport everything from one side of the river to the other? Formulate it in terms of a Planning Domain Definition Language (PDDL).
- Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
- Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
- Implement k-nearest neighbors classification using python.
- Implement linear regression using python.
- Implement the naïve Bayesian classifier for a sample training dataset. Compute the accuracy of the classifier, considering few testdata sets.

#### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply AI search Models and Generic Search strategies for problem solving.                                                                                      | Apply        | 1          | 1, 3             |
| 2      | Inspect and analyze Logic for representing Knowledge and Reasoning of AI systems and Conduct investigation and implement project using AI learning techniques. | Analyze      | 2          | 2                |
| 3      | Apply and evaluate the searching strategies to achieve the goal for a given situation.                                                                         | Apply        | 1          | 6                |
| 4      | Design different learning algorithms for improving the performance of AI systems.                                                                              | Apply        | 2          | 4                |

#### TEXT BOOK:

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", Fourth Edition, Pearson Education, 2010.

#### REFERENCE BOOKS:

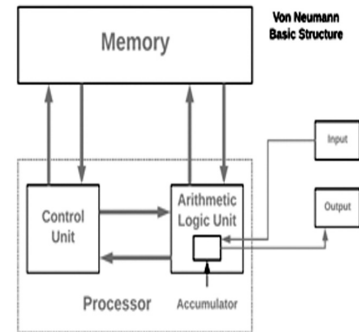
1. David Poole, Alan Mackworth, Randy Goebel, "Computational Intelligence : a logical approach", Oxford University Press. 2018.
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problemsolving", Fourth Edition, Pearson Education, 2010.
3. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers, 2013.



# 22AM202 DIGITAL LOGIC AND COMPUTER ORGANIZATION

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 0 | 3 |



Source: computer organization & architecture and digital logic design course - Bing images

**PREREQUISITE KNOWLEDGE:** Algebra.

## COURSE DESCRIPTION AND OBJECTIVES:

The emphasis is on understanding the design of computer and its components. This course introduces the basic knowledge on number systems, design of combinational and sequential circuits. The course mainly focuses on micro operations, memory organizations and input output organization.

## MODULE-1

### UNIT-1

**8L+8T+0P=16 Hours**

#### NUMBER SYSTEMS AND BOOLEAN ALGEBRA:

**Number Systems:** Binary Numbers, Number base Conversions, Complements, Binary codes.

**Boolean Algebra:** Fundamental concepts of Boolean algebra basic theorems and properties of Boolean Functions.

**Gate-Level Minimization:** Canonical and standard forms - SOP and POS forms, Digital Logic gates, The map method – two, three, four variable K map; POS and SOP simplification; Don't care conditions;

### UNIT-2

**8L+8T+0P=16 Hours**

#### COMBINATIONAL LOGIC CIRCUITS:

**Combinational Logic Circuits:** combinational circuits analysis, design procedure; Half adder, Full adder, Half subtractor, Full subtractor, Binary adder/subtractor; Binary Incrementor, Decimal Adder, Binary multiplier; Magnitude comparator; Decoders; Encoders; Multiplexers; De-Multiplexer. Synchronous Sequential Logic: Sequential circuits; storage elements - latches, flip flops; Flipflop conversion; counters.

#### PRACTICES:

- Design a combinational circuit with three inputs and one output. The output is 1 when the binary value of the inputs is less than 3. The output is 0 otherwise.
- Design a combinational circuit with three inputs x, y, z and three outputs A, B, C. When the binary inputs is 0, 1, 2 or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7 then the binary output is one less than the input.
- Design a code converter that converts a decimal digit from the 8, 4, -2, -1 code to BCD.
- Implement a Full – Adder using 4 X 1 multiplexer.
- Design a 16 X 1 Multiplexer with five 4 X 1 multiplexers.
- Design a 5-to-32line decoder with four 3-to-8-line decoders with enable and one 2-to-4-line decoder.
- Design a JK flip-flop using a D flip-flop.
- Design a sequential circuit with two D flip-flops A and B and, one input x. When x=0, the state of the circuit remains same. When x = 1, the circuit goes through the state transitions from 00 to 01, 01 to 11, 11 to 10, 10 back to 00 and repeats until the clock pulse is active.
- Design a 4-bit binary synchronous counter using T flip-flop.

**SKILLS:**

- ✓ *Digital circuit Design.*
- ✓ *Number system Conversion.*
- ✓ *Analyse the principles of computer architecture.*
- ✓ *Design digital circuitry for implementing different operations.*
- ✓ *Identify the types of memories and their uses*
- ✓ *Study various data transfer mechanisms in digital computer and I/O*

**MODULE-2****UNIT-1****8L+8T+0P=16 Hours****INTRODUCTION:**

**Introduction to Computer Organization and Architecture:** Organization and architecture, structure and function; RTL - Register transfer language, register transfer, bus and memory transfers.

**Micro Operations:** Arithmetic micro operations; Logic micro operations; Shift micro operations; Arithmetic logic shift unit.

**Basic Computer Organization and Design:** Instruction codes; Computer registers; Computer instructions; Instruction cycle; Memory reference instructions; Register reference instructions, Input/Output-Interrupt.

**UNIT-2****8L+8T+0P=16 Hours****MEMORY AND INPUT-OUTPUT ORGANIZATION:**

**Memory Organization:** Memory hierarchy; Main memory; Associative memory; Cache memory; Virtual memory.

**Input- Output Organization:** Asynchronous data transfer; Modes of transfer; Priority interrupt; Direct memory access.

**PRACTICES:**

- Design a Common bus system for eight registers with eight bits each using multiplexers.
- Design a Common bus system for four registers with four bits each using three state gate buffers.
- A digital computer has a common bus system for 16 registers of 32 bits each. The bus is constructed with multiplexers
  - How many selection inputs are there in each multiplexer?
  - What size of the multiplexers are needed?
  - How many multiplexers are there in the bus?
- Perform arithmetic operations  $(+42) + (-13)$  and  $(-42) - (-13)$  in binary using signed 2's complement representation for negative numbers.
- Design a 4-bit combinational decremter circuit using 4 full adder circuits.
- Register A holds the 8-bit binary 11011001. Determine the B operand and the logic micro operation to be performed in order to change the value in A to :
  - a. 01101101
  - b. 11111101
- An 8-bit register contains the binary value 10011100. What is the register value after an arithmetic shift right? Starting from the initial number 10011 100, determine the register value after an arithmetic shift left, and state whether there is an overflow.
- Starting from an initial value of  $R = 11011101$ , determine the sequence of binary values in R after a logical shift-left, followed by a circular shift-right, followed by a logical shift-right and a circular shift-left.
- Design arithmetic logic shift unit that performs different operations on 4 bits.
- The content of PC in the basic computer is 3AF (all numbers are in hexadecimal). The content of AC is 7EC3. The content of memory at address 3AF is 932E. The content of memory at address 32E is 09AC. The content of memory at address 9AC is 8B9F.
  - What is the instruction that will be fetched and executed next?
  - Show the binary operation that will be performed in the AC when the instruction is executed.
  - Give the contents of registers PC, AR, DR, AC, and IR in hexadecimal and the values of E, I, and the sequence counter SC in binary at the end of the instruction cycle.
- How many characters per second can be transmitted over a 1200-baud line in each of the following modes? (Assume a character code of eight bits.)
  - Synchronous serial transmission.
  - Asynchronous serial transmission with two stop bits.
  - Asynchronous serial transmission with one stop bit.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply the knowledge of digital logic concepts to optimize digital circuits and Boolean algebra rules & Karnaugh map method for reducing the Boolean functions. | Apply        | 1          | 1                |
| 2      | Analyse Combinational and sequential digital circuits for the given problem statement.                                                                         | Analyse      | 1          | 2                |
| 3      | Design of Combinational and sequential digital circuits for the given problem statement and improve the performance by reducing the complexities.              | Creating     | 1          | 3                |
| 4      | Demonstrate computer architecture concepts related to the design of modern processor, memory, and I/O systems.                                                 | Analyse      | 2          | 1, 2             |
| 5      | Evaluate the performance of a processor and memory in terms of speed, size and cost.                                                                           | Evaluation   | 2          | 1, 2             |

**TEXT BOOKS:**

1. M Morris Mano and Michael D. Ciletti, "Digital Design", 5th edition, Pearson Education, 2013.
2. M.Moris Mano, "Computer Systems Architecture", 3rd edition, Pearson/Prentice Hall India, 2007.

**REFERENCE BOOKS:**

1. John F.Wakerly, "Digital Design Principles and Practices", Third Edition, Pearson/PHI, 2015.
2. Charles H.Roth. "Fundamentals of Logic Design", 6th Edition, Thomson Learning, 2013.
3. John.M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2006.
4. Carl Hamacher, ZvonkoVranesic and SafwatZaky, "Computer Organization", 5th edition, Tata McGraw Hill, 2007.
5. William Stallings, "Computer Organization and Architecture", 6th edition, Pearson/Prentice Hall India, 2007.

# 22CS201 DATABASE MANAGEMENT SYSTEMS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |



DBMS

What Is Database Management System ?  
www.learncomputerscienceonline.com

Source: <https://www.youtube.com/watch?v=IDpB9zF8LBw>

**PREREQUISITE KNOWLEDGE:** Discrete Mathematical Structures.

## COURSE DESCRIPTION AND OBJECTIVES:

This course presents an introduction to database management systems with an emphasis on how to organize, maintain and retrieve data efficiently from a relational database. It also focuses on requirements gathering and conceptual, logical, physical database design. The objective of the course is to enable the student to understand database design, expressing queries using SQL, query optimization and transaction processing.

## MODULE-1

### UNIT-1

**10L+6T+4P=20 Hours**

#### DATABASE SYSTEM CONCEPTS:

**Databases And Database Users:** Introduction; Characteristics of the database approach; Actors on the scene; Advantages of using DBMS approach.

**Database System Concepts and Architecture:** Data models, Schemas and instances; Three-Schema architecture and data Independence; Database languages and interfaces; The database system environment; Centralized and Client-Server architectures for DBMS.

**Conceptual Data Modeling and Database Design:** Entity types, Entity sets, Attributes and keys; Relationship types, Relationship sets, Roles and structural constraints; Weak entity types; Relationship types.

### UNIT-2

**6L+10T+12P=28 Hours**

#### RELATIONAL DATABASE DESIGN:

**Relational Database Design by ER-To-Relational Mapping:** Relational Database design using ER-to-Relational mapping.

**The Relational Data Model and Relational Database Constraints:** Relational model concepts; Relational model constraints and Relational database schemas.

**Relational Algebra:** Unary relational operations - SELECT and PROJECT; Relational algebra operations from set theory; Binary relational operations - JOIN and DIVISION.

**SQL:** SQL data definition and data types; specifying constraints in SQL, Basic retrieval queries in SQL; INSERT, DELETE, and UPDATE statements in SQL.

#### PRACTICES:

- Design ER Model for various real time database applications.
- Development of Relational Database schemas for Company/Student/Sailors/ using DDL constructs of SQL.
- Apply various DML Commands such as select, insert, update etc. of SQL on Relational Database.
- Design of Relational Database schemas by specifying different types of Constraints.
- Apply various Relational Database operators (Arithmetic, Logical & comparison) and string-matching constructs of SQL.
- Expressing queries using Aggregate Functions of SQL on Relational Database.
- Queries on Relational Database using GROUP BY, HAVING and ORDER BY clauses of SQL.

**MODULE-2****UNIT-1****8L+8T+12P=28 Hours****NORMALIZATION:**

**Complex Queries, Triggers, Views:** More complex SQL retrieval queries; Specifying constraints as assertions and actions as triggers; Views (virtual tables) in PL/SQL.

**Basics of Functional Dependencies and Normalization for Relational Databases:** Informal design guidelines for relation schemas; Functional dependencies-inference rules, equivalence and minimal cover; Normal forms based on primary keys; Boyce-Codd normal form; Properties of relational decompositions, multi valued dependency, join dependencies.

**UNIT-2****8L+8T+4P=20 Hours****TRANSACTION PROCESSING:**

**Introduction To Transaction Processing Concepts and Theory:** Introduction to transaction processing; Transaction and system concepts; Desirable properties of transactions; Characterizing schedules based on serializability.

**Concurrency Control Techniques:** Two-phase locking techniques for concurrency control, concurrency control based on timestamp ordering.

**Database Recovery Techniques:** Recovery concepts; Shadow paging; The ARIES recovery algorithm.

**Indexing Structures for Files and Physical Database Design:** Single level and multi-Level indexing; Dynamic multi-level indexing using B-trees and B+ trees.

**PRACTICES:**

- Design and Development of company database and expressing Nested queries using SQL.
- Design and Development of student database and specifying queries using set operations.
- Design and Development of sailor's database and specifying queries using different types of JOINS.
- Implementation of PL/SQL programs with Control Structures.
- Implementation of PL/SQL programs with Procedures.
- Implementation of PL/SQL programs with Function.
- Implementation of PL/SQL programs with Triggers.
- Creation and dropping of VIEWS.
- Relation R has eight attributes ABCDEFGH. Fields of R contain only atomic values.  $F = \{CH \rightarrow G, A \rightarrow BC, B \rightarrow CFH, E \rightarrow A, F \rightarrow EG\}$  is a set of functional dependencies (FDs) so that F+ is exactly the set of FDs that hold for R. How many candidate keys does the relation R have?
- Apply various DCL and TCL constructs of SQL on Relational Database.

**SKILLS:**

- ✓ Develop E-R model for real life applications.
- ✓ Design of relational databases for real world applications.
- ✓ Devise queries using relational algebra and SQL.
- ✓ Analyze transaction processing, concurrency control and recovery techniques

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                               | Blooms Level | Module No. | Mapping with POs |
|--------|---------------------------------------------------------------|--------------|------------|------------------|
| 1      | Develop an E-R model for real life applications.              | Apply        | 1          | 1, 10            |
| 2      | Design and normalize databases for real time applications.    | Create       | 1          | 1, 3             |
| 3      | Devise queries using Relational Algebra and SQL.              | Analyze      | 2          | 2                |
| 4      | Express queries using database tools like Oracle, DB2, MYSQL. | Apply        | 2          | 5, 10            |

**TEXT BOOKS:**

1. Ramez, Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", 7th Edition, Pearson Education, 2016.
2. Raghu Rama Krishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition, Tata McGraw Hill, 2013.

**REFERENCE BOOKS:**

1. Abraham Silberschatz, Henry F.Korth and S.Sudarshan, "Database System Concepts", 7th edition, Tata Mc Graw Hill,2019.
2. Allen G. Taylor "Database Development for Dummies" 1st Edition, 2011
3. C. J. Date "Introduction to Database Systems" 7th Edition, Addison Wesley, 2003.

**22CT201 ENVIRONMENTAL STUDIES**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 1 | 1 | 0 | 1 |

**PREREQUISITE KNOWLEDGE:** General awareness regarding environmental problems and importance of environmental protection.

**COURSE DESCRIPTION AND OBJECTIVES:**

It is a multidisciplinary subject where different aspects of society and environment are dealt using a holistic approach. It is evolving to be the education for sustainable and ethical development both at a local and global level. It helps to prepare the next generation for planning appropriate strategies to address environmental issues. It identifies and creates solutions that conserve to manage ecosystem and biodiversity and helps to eliminate pollutants, toxicants, preserve air, water and soil quality. Environmental education recognizes impacts of global issues, enhances the public awareness and helps to take decisions towards environmentally responsible actions.

**MODULE-1****UNIT-1****4L+4T+0P=8 Hours****INTRODUCTION TO ENVIRONMENT:****NATURAL RESOURCES, ECOSYSTEMS AND BIODIVERSITY:**

Environment and sustainable development; Natural resources- forest, water, energy and land resources; Ecosystem – basic structural components, function and interactions in ecosystem, ecological succession.

**UNIT-2****4L+4T+0P=8 Hours****BIODIVERSITY AND CONSERVATION:**

Introduction to biodiversity, types of biodiversity- species, genetic and ecosystem diversity; Threats to biodiversity - natural and anthropogenic, species extinctions, man wildlife conflicts; Biodiversity conservation - principles and strategies; in-situ and ex-situ conservation.

**PRACTICES:**

- Visit to a Biogas plant, Solar Power plant.
- Visit to a local area: river / pond / lake / forest / grassland / hill / mountain and study of different types of ecosystems, biodiversity study and documentation (herbarium sheet preparation).
- Set up an aquarium.
- Case study: Renewable energy use.

**MODULE-2****UNIT-1****4L+4T+0P=8 Hours****ENVIRONMENTAL POLLUTION AND CLIMATE CHANGE:**

Air, water, soil, radioactive and noise pollution; Study of different pollutants (SO<sub>x</sub>, NO<sub>x</sub>, PAN, PAH etc.); Toxicity study; Climate change - greenhouse effect, acid rain, ozone layer depletion.

**UNIT-2****4L+4T+0P=8 Hours****POLLUTION CONTROL DEVICES AND WASTEWATER TREATMENT TECHNOLOGIES:**

Air pollution control devices - Gravitational settling chambers, cyclonic separators, electrostatic precipitators, fabric filters and bio filters, Wastewater management.



Source: Biogas plant at VFSTR

**SKILLS:**

- ✓ Create a biodiversity map of any habitat/ ecosystem.
- ✓ Strategize different ways of using renewable energy resources.
- ✓ Design novel strategies and approaches for pollution control and waste management

**PRACTICES:**

- Visit to a sewage treatment plant and wastewater analysis.
- Case study: Recycling Technologies.
- Case study: Effects of contaminants on microorganisms.
- Report writing: 12 principles of green chemistry for environmental sustainability.
- Report writing: Environmental Impact Analysis, Local Disaster Management Plan.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                            | Blooms Level | Module No. | Mapping with POs          |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|---------------------------|
| 1      | Apply the basic concepts of sustainable development, natural resource utilization and ecology for the purpose of environmental protection. | Apply        | 1          | 1, 6, 7, 9, 10, 11, 12    |
| 2      | Design remediation technologies for their abatement                                                                                        | Apply        | 2          | 1, 3, 6, 7, 9, 10, 11, 12 |
| 3      | Analyze the biodiversity of different ecosystems and formulate various conservation approaches                                             | Analyze      | 1          | 1, 7, 8, 9, 10, 11, 12    |
| 4      | Analyze the presence of various environmental pollutants.                                                                                  | Analyze      | 2          | 1, 6, 7, 9, 10, 11, 12    |
| 5      | Recommend various waste management approaches and their implementation strategies.                                                         | Evaluate     | 2          | 1, 2, 7, 8, 9, 10, 11, 12 |

**TEXT BOOKS:**

1. A. Kaushik and C. P. Kaushik, "Perspectives in Environmental Studies", New Age International Publishers, 5th Edition, 2016.
2. Y. Anjaneyulu, "Introduction to Environmental Science", B. S. Publications, 2015.

**REFERENCE BOOKS:**

1. B. Joseph, "Environmental Studies", Mc Graw Hill Education, 2nd Edition, 2015.
2. S. Subash Chandra, "Environmental Science", New Central Book Agency, 2011.
3. M. Basu and S. Xavier, "Fundamentals of Environmental Studies", Cambridge University Press, 2016.
4. K. Muktanti, "A Textbook of Environmental Studies", S. Chand Company Ltd., 2009.
5. M. Anji Reddy, "A Textbook of Environmental Science and Technology", B. S. Publications, 2008.



# 22TP203 ADVANCED CODING COMPETENCY

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 2 | 1 |

COMPETITIVE  
PROGRAMMING

**PREREQUISITE KNOWLEDGE:** Programming in C, Data Structures.

## COURSE DESCRIPTION AND OBJECTIVES:

This course helps to understand the impact of the choice of data structures and design strategies to solve the problem in an efficient manner. This course also provides the understanding of advanced graph applications and also throw light in tractable intractable problems.

## MODULE-1

### UNIT-1

0L+0T+8P=8 Hours

### STACKS, QUEUES AND SINGLE LINKED LISTS:

#### PRACTICES:

#### PROBLEMS ON STACKS & QUEUES:

- Check if given stack of integers are consecutive or not (could be ascending or descending).
- Find the maximum sum in a sliding window using queues.
- Given a queue of integers, rearrange the elements by interleaving the first half with the second half.
- Given an integer k and a queue of integers, reverse the order of the first k elements of the queue.
- Given a maze in the form of a rectangular matrix filled with O, X or M where O represents an open cell, X represents a blocked cell and M represents landmines, find the shortest distance of every open cell in the maze from its nearest mine.
- For a given parenthesis expression, check whether it is balanced parenthesis or not.
- Reverse a number using stack.
- You are given a string s consisting of lowercase English letters. A duplicate removal consists of choosing two adjacent and equal letters and removing them. We repeatedly make duplicate removals on s until we no longer can.
- Find first Unique character in a string (Queue).
- Implement Tower of Hanoi problem.

#### PROBLEMS ON LINKED LISTS:

- Given a random pointer to a random node in a singly linked list, clone the list.
- Given a list rotate the list to the right by k places.
- Remove duplicates from a sorted list.
- Find fractional node in a singly linked list.
- Sort a linked list using constant space complexity.
- Delete a node in start, middle, end of Singly linked list.
- Add a node in start, middle, end of Singly linked list.
- Find whether given single linked list is circular or not.
- Arrange a singly linked list in Descending order.
- Addition of two numbers using Singly Linked List.

Source: <https://www.geeksforgeeks.org/best-way-to-start-with-competitive-programming-geeksforgeeks-cp-live-course/>

**SKILLS:**

- ✓ Experienced to Store data and various types of data to handle.
- ✓ Ordering and sorting of data.
- ✓ Indexing and Searching of required data from large data sequences.
- ✓ Exposed to various characteristics such as Linear or non-linear, Homogeneous or heterogeneous and Static and Dynamic

**UNIT-2****0L+0T+8P=8 Hours****DOUBLY LINKED LISTS, CIRCULAR LINKED LISTS:****PRACTICES:****PROBLEMS ON DOUBLE LINKED LISTS AND CIRCULAR LINKED LISTS:**

- Implement a clockwise rotation of a doubly linked list by N places.
- Count triplets in a sorted doubly linked list whose product is equal to a given value x.
- Find the product of all prime nodes in a doubly linked list.
- Find the count of common nodes in two doubly linked lists.
- Find pairs with given product in a sorted doubly linked list.
- Delete all the even nodes of a circular singly linked list.
- Count nodes in a circular linked list.
- Delete all prime nodes from a circular singly linked list.
- Exchange first and last nodes in a circular linked list.
- Reverse a doubly circular linked list.
- Linear search using a stack of incomplete sub problems.
- 1 2 3 4 5 6 in stack S is push X is pop, SSSSXXSSSXXX.
- Recursively remove all adjacent duplicates.
- Check if a given singly linked list is a palindrome using stack.
- Convert a multilevel singly linked list to a singly linked list.
- Remove duplicates from an unsorted doubly linked list.
- Sort a doubly linked list using insertion sort.
- Check if a doubly linked list of characters is palindrome or not.
- Swap Kth node from beginning with Kth node from end in a Double Linked List.
- Convert a Binary Tree into Double Linked List.

**MODULE-2****UNIT-1****0L+0T+8P=8 Hours****TREES:****PRACTICES:****PROBLEMS ON TREES:**

- Given a sorted doubly linked list, convert it into a balanced BST.
- Given a singly linked list with data in the ascending order, convert it into a height balanced BST.
- Print the leaf to root path for every leaf node in a binary tree.
- Write a function to implement the reversed level order traversal of a binary tree.
- Truncate a given binary tree to remove nodes that lie on a path having sum less than K.
- Find the vertical sum in a given binary tree.
- Delete minimum & Maximum element from a BST.
- Implement Inorder, preorder and postorder tree traversal techniques.
- Print Kth largest element in a BST.
- Implement Zig-Zag tree traversal.

**UNIT-2****0L+0T+8P=8 Hours****GRAPHS:****PRACTICES:****PROBLEMS ON GRAPHS:**

- Given a directed acyclic graph, determine whether there is a path that visits every vertex exactly once.
- Reverse a directed graph such that each edge from v to w is replaced by an edge from w to v.
- Find the shortest path in a graph that visits each vertex at least once, starting and ending at the same vertex.
- Find the minimum number of throws required to win a snake and ladder game.
- Implement DFS of a Graph.
- Implement BFS of a Graph.
- Detect whether a cycle is present in an undirected graph.
- Detect cycle in a Directed Graph.
- Find Shortest Distance to goal node from root node in a graph.
- Find no. of nodes in Kth level of a Graph.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                          | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply various data structures to solve a different algorithm.                            | Apply        | 1, 2       | 1                |
| 2      | Investigate the various data structures to solve a given problem in an efficient manner. | Analyse      | 1, 2       | 2                |
| 3      | Design and implement an appropriate hashing function for an application.                 | Create       | 1, 2       | 4                |

**TEXT BOOKS:**

1. Reema Thareja, "Data Structures Using C", 2nd Edition, Oxford University Press, 2014.
2. Seymour Lipschutz, "Data Structures with C", 1st Edition, McGraw Hill Education, 2017.

**REFERENCE BOOKS:**

1. Ellis Horowitz and Sartaj Sahni, "Fundamentals of Data Structures", illustrated edition, Computer Science Press, 2006.
2. Richard F. Gilberg and Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", 2nd Edition, CENAGE Learning, 2005.
3. R G Dromey and Pearson, "How to solve it by Computer", 2nd edition, Impression edition, 1998.

# 22TP204 PROFESSIONAL COMMUNICATION

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 2 | 1 |

**PREREQUISITE KNOWLEDGE:** High School-level English.

## COURSE DESCRIPTION AND OBJECTIVES:

To improve the overall professional communication skills (LSRW) of students and prepare them for their profession as engineers and managers. To provide them exposure to conventions of corporate communication and training them on how to function in the business world.

### MODULE-1

#### UNIT-1

0L+0T+8P=8 Hours

#### **BASICS OF BUSINESS WRITING SKILLS, PRACTICING BUSINESS CORRESPONDENCE AND REPORT WRITING:**

**Business English Vocabulary:** Glossary of most commonly used words (formal and informal usage).

**Elements of Technical Writing:** Sentence structure, reducing verbosity, arranging ideas logically, building coherence, cohesive devices and transitional words.

**Mechanics of Writing:** elementary rules of grammar, choice of diction, elementary principles of composition, matters of form, punctuation, conventions of business communication, language and professional tone, code of conduct (not sending illegal, offensive, disparaging personal remarks or comments) in written business communication.

**Business Correspondence:** E-mail: nature and scope, e-mail etiquette, clear call for action, common errors in composing e-mails, office communication such as meeting agenda and notice, circular and memo.

**Letter-Writing:** Formal and informal letters, structure of formal letters, expressions of salutations, different types of letters [such as sales letter, complaint letter, response to the complaint letter (dispute resolution), letter of permission, letter of enquiring, claim letter – letter of apology etc], introductory and concluding paragraphs and clear call for action.

**Professional Proposal/Report:** Differentiating proposals and reports, Drafting formal business proposals, types of reports such as factual reports, feasibility reports and survey reports, parts of a report (such as title page, declaration, acknowledgements, table of contents, abstract, introduction, findings, conclusion and recommendations).

**New Age Corporate Communication Media:** Importance of social media communication and Etiquettes, form and structure, sharing texts through Twitter, Whatsapp, instgram etc.

#### UNIT-2

0L+0T+8P=8 Hours

#### **PRACTICING COMMUNICATIVE LANGUAGE IN VARIOUS PROFESSIONAL CONTEXTS:**

**Speaking:** Speaking in business context, assertiveness, politeness, making requests, queries and questions, negotiations, asking for information, offering suggestions, conflict resolution, contacting clients, initiating, addressing delegates (in public), delivering the presentation effectively, telephone etiquettes, delivering seminar/proposal/report effectively, team meeting etiquettes (face to face and conference call), making effective one minute presentations(JAM) and participating in Group Discussions.

#### **PRACTICES:**

- Basic grammar practice, framing paragraphs on topics allocated, paraphrasing an article or a video in your own words, finding topic sentences in newspaper articles, finding out new words from a professional viewpoint and understanding the meaning and its usage.

- Perusing samples of well-prepared business emails, memo, letter writing and short proposals and reports, students will draft business correspondence writing tasks and different proposals/ reports on topics assigned.
- Watching videos/listening to audios of business presentations, classroom activities of team and individual presentations, using PPTs, mock exercises for BEC speaking, agreeing, disagreeing politely, developing content, extended speaking in Group Discussion(s).

## MODULE-2

### UNIT-1

0L+0T+8P=8 Hours

#### READING AND COMPREHENDING BUSINESS DOCUMENTS:

**Reading:** Reading and comprehending business documents, learning business register, regularizing the habit of reading business news, suitable vocabulary, skimming and scanning a text for effective and speedy reading and dealing with ideas from different sectors of corporate world in different business contexts.

### UNIT-2

0L+0T+8P=8 Hours

#### IMPARTING AND PRACTICING LISTENING SKILLS:

**Listening:** Specific information in business context, listening to telephonic conversations / messages and understanding the correct intended meaning, understanding the questions asked in interviews or in professional settings, summarizing speaker's opinion or suggestion, enable active listening.

#### PRACTICES:

- Hand-outs; matching the statements with texts, finding missing appropriate sentence in the text from multiple choices, using right vocabulary as per the given context and editing a paragraph.
- Working out BEC/TOEFL/IELTS listening exercises with hand-outs; matching the statements with texts, finding missing appropriate sentence in the text from multiple choice- multiple choices, using right vocabulary in context-editing a paragraph, listening to a long conversation such as an interview and answer MCQ s based upon listening.

#### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                               | Blooms Level | Module No. | Mapping with POs |
|--------|---------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Possess comprehensive skills in listening and reading business texts in formal context.                       | Apply        | 2          | 7                |
| 2      | Communicate effectively both in their academic as well as professional environment.                           | Apply        | 1, 2       | 10               |
| 3      | Clear grasp on the register of business language.                                                             | Analyze      | 1          | 8                |
| 4      | Possess the ability to write business reports and proposals clearly and precisely to succeed in their future. | Create       | 1          | 12               |
| 5      | Make effective presentations and participate in formal context.                                               | Create       | 2          | 10               |

#### TEXT BOOK:

1. S. Schnurr, "Exploring Professional Communication: Language in Action", London: Routledge, 2013.

#### REFERENCE BOOKS:

1. Brook Hart Guy, "Cambridge English Business Bench Mark: Upper Intermediate", 2nd Edition: CUP, 2014.
2. Cambridge University Publication, "Cambridge: BEC VANTAGE Practice Papers", CUP, 2002.
3. J. Seely, "The Oxford Guide to Effective Writing and Speaking", Oxford University Press, 2005.

#### SKILLS:

- ✓ To enhance listening and spoken abilities of students needed for professional and social success in interpersonal situations, group interactions, and personal and professional presentations.
- ✓ Know and practice specific functions and vocabulary in a business context.
- ✓ Produce short business reports, proposals and correspondence.
- ✓ Write various business documents through reading techniques.

# 22AM203 DESIGN AND ANALYSIS OF ALGORITHMS USING PYTHON

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Programming for problem solving, Discrete Mathematical Structures, Data Structures.

## COURSE DESCRIPTION AND OBJECTIVES:

This course offers the basic knowledge required to analyze the asymptotic performance of algorithms. In addition, this course provides the knowledge required to solve different problems using suitable design strategies such as the greedy method, divide and conquer, dynamic programming, backtracking and branch & bound. This course helps to understand the impact of the choice of data structures and algorithm design strategies on the performance. This course also provides the understanding of advanced graph applications and throws light on tractable and intractable problems.

## MODULE-1

### UNIT-1

6L+6T+6P=18 Hours

#### INTRODUCTION:

**Introduction:** Algorithm, Pseudo-code for expressing algorithms, Performance analysis – space and time complexity; Asymptotic notation - big oh notation, Omega notation, Theta notation and little oh notation; Analysis of recursive algorithms through recurrence relations- substitution method, Recursion tree method, Masters Theorem.

**Disjoint sets:** Disjoint set operations, Union and find algorithms.

### UNIT-2

10L+10T+10P=30 Hours

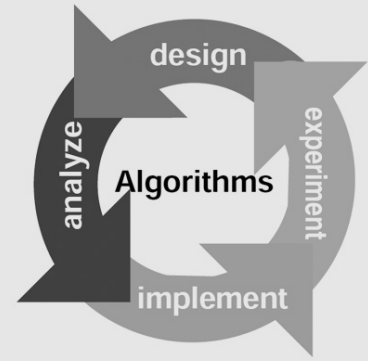
#### DIVIDE & CONQUER AND GREEDY METHOD:

**Divide and Conquer:** General method, Applications - Binary search, Quick sort, Merge sort and Strassen's matrix multiplication.

**Greedy Method:** Applications - job sequencing with deadlines, Knapsack problem, Minimum cost spanning trees.

#### PRACTICES:

- Sort a given set of elements using the following methods and determine the time required to sort the elements. Repeat the experiment for different values of  $n$ , the number of elements in the list to be sorted and plot a graph of the time taken versus  $n$  inputs. The elements can be read from a file or can be generated using the random number generator.
  - a. Quick sort
  - b. Merge sort
- Search for a given set of elements using the following methods and determine the time required to search the given element. Repeat the experiment for different values of  $n$ , the number of elements in the list to be sorted and plot a graph of the time taken versus no. of elements. The elements can be read from a file or can be generated using the random number generator.
  - a. Linear Search
  - b. Binary Search
- Implement the following using divide and conquer approach.
  - To multiply two given square matrices.
  - To multiply two given square matrices using Strassen's matrix multiplication.
- Design the Algorithm to solve Job sequencing with deadlines problem and Analyze its time complexity. Implement the above algorithm using Greedy method.



Source: <https://www.facebook.com/Design-and-Analysis-of-Algorithms-1553902878155564/>

- Design the Algorithm to solve fractional Knapsack problem using Greedy method. Analyze the time complexity and implement the above algorithm.
- Design the Algorithm to find minimum spanning tree and its cost for an undirected graph. Analyze the time complexity and implement the above algorithm.

## MODULE-2

### UNIT-1

10L+10T+10P=30 Hours

#### DYNAMIC PROGRAMMING AND BACKTRACKING:

**Dynamic Programming:** General method, Applications - optimal binary search trees, Matrix chain multiplication, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem.

**Backtracking:** General method, Applications - N-Queen problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles.

### UNIT-2

6L+6T+6P=18 Hours

#### BRANCH & BOUND P, NP, NP - HARD AND NP-COMPLETE:

**Branch and Bound:** General method, Applications- Travelling sales person problem, 0/1 knapsack problem using LC branch and bound solution and FIFO branch and bound solution.

**P, NP, NP - HARD and NP-Complete:** Basic Concepts - Non-Deterministic Algorithms - The Classes NP-Hard and NP Complete- NP Hard Problems- Clique Decision Problem-Cook's Theorem.

#### PRACTICES:

- Design the Algorithm to find all pairs shortest path problem by using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal binary search tree and its cost by using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal order of matrix chain multiplication and its cost using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal route for travelling sales person problem and its cost by using dynamic Programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to solve N-queens problem by using backtracking approach and Analyze its time complexity. Implement the above algorithm.
- Design the Algorithm to solve sum of subsets problem using backtracking approach and Analyze its time complexity. Implement the above algorithm.
- Design the Algorithm to solve 0/1 Knapsack problem using Branch and Bound method. Analyze the time complexity and Implement the above algorithm.

#### SKILLS:

- ✓ Analyze the given algorithm concerning space and time complexities and compare it with other algorithms.
- ✓ Develop algorithms for solving problems using divide and conquer, greedy, dynamic programming, backtracking and branch & bound techniques.
- ✓ Application of existing design strategies to solve real-world problems.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                 | Blooms Level | Module No. | Mapping with POs |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Analyze the efficiency of a given algorithm using time and space complexity theory. Understanding algorithmic design strategy like divide and conquer approach. | Analyze      | 1          | 1, 2, 12         |
| 2      | Apply greedy algorithm Strategy for suit able problems and argue the correctness of such algorithms with respect to the global optimization.                    | Apply        | 1          | 1, 2, 3, 5, 12   |
| 3      | Understand the dynamic programming paradigm and identify the kind of problem best suited to solve using dynamic programming.                                    | Apply        | 2          | 1, 2, 3, 5, 12   |
| 4      | Compare and contrast the design principles of branch and bound with backtracking strategy.                                                                      | Apply        | 2          | 1, 2, 3, 5, 12   |
| 5      | Investigate computational complexity of different class of problems.                                                                                            | Analyze      | 2          | 1, 2, 4,12       |

**TEXT BOOKS:**

1. Ellis Horowitz, SatrajSahni and Rajasekharan, "Fundamentals of Computer Algorithms", 2nd edition, Galgotia publications, 2006.
2. Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithm", 2nd Edition, MIT press Ltd., 2014.

**REFERENCE BOOKS:**

1. Anony Levitin, "Introduction to Design and Analysis of Algorithms", 3rd Edition, Pearson Education, 2016.
2. Donald E. Knuth, "The Art of Computer Programming", 2nd Edition, Addison Wesley Publishing Company, 1998.
3. Ronald L. Graham, Donald E. Knuth and Oren Patashnik, "Concrete Mathematics", 2nd Edition, Addison wesley Publishing Company, 1998.
4. Dasgupta, Papadimitriou and Vazirani, "Algorithms", 1st Edition, McGraw-Hill publishers, 2008.
5. Weiss, "Data Structures and Algorithm Analysis", 1st Edition, Addison-Wesley Publishing Company, 2016.



**22AM204 MACHINE LEARNING**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Probability & Linear Algebra, Python language.

**COURSE DESCRIPTION AND OBJECTIVES:**

This course provides a broad introduction to various machine learning concepts including Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks) and Unsupervised learning (clustering, dimensionality reduction) methods. Students will get an understanding of various challenges of Machine Learning and will be able to decide on model complexity. Numerous case studies introduced in this course allow the students to apply machine-learning algorithms in computer vision, medical imaging, audio, and text domains. Laboratory experiments of this course will introduce students to advanced Machine Learning Python libraries such as Scikit-Learn, Matplotlib, and many other recent ML-related APIs. The course is designed such that the students get enough hands-on experience with a major focus on the practical implementation of theoretical concepts.

**MODULE-1****UNIT-1****14L+0T+8P=22 Hours****INTRODUCTION:**

What is machine learning? Machine learning applications; Types of Learning: Supervised learning; Un-supervised learning; Reinforcement learning.

**Model Training Essentials:** Re-sampling methods: Bias–Variance Trade-off. Hypothesis Testing and Variable Selection, Sub sampling and Upsampling, SMOTE; Cross Validation (validation set, Leave-One-Cut (LOO), k-fold strategies) and bootstrap; Evaluation measures-Error functions, Confusion Matrix, Accuracy, Precision and Recall, F1 Score.

**Regression Analysis:** Linear Regression, Simple and Multiple Linear Regression, Polynomial Regression, Logistic Regression, Multi nominal Regression. Ordinary Least Squares Method, Model Shrinkage-Ridge, and LASSO regression.

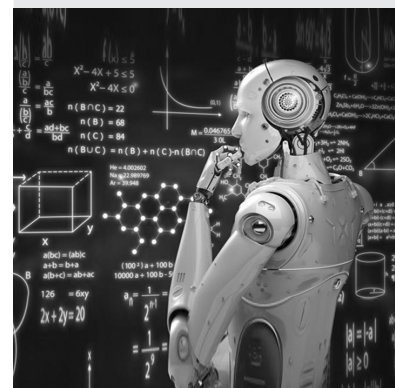
**UNIT-2****10L+0T+8P=18 Hours****FEATURE SELECTION:**

**Feature Selection Strategies:** Problem statement and Uses, Filter methods, Wrapper methods, Embedded methods. Branch and bound algorithm, Sequential forward/backward selection algorithms.

**Dimensionality Reduction:** Singular value decomposition, matrix factorization, Linear discriminant analysis, Principal components analysis.

**PRACTICES:**

- Apply the following tasks to any given dataset:
  - a. Load and visualize data
  - b. Check out and replace missing values
  - c. Encode the Categorical data
  - d. Splitting the dataset into Training and Test set
  - e. Splitting the dataset into k-folds
  - f. Feature scaling
- House price prediction:
  - a. Create a model that predicts a continuous value (price) from input features square footage, number of bedrooms and bathrooms.).
  - b. Implement a univariate Model using Least Squares and plot best-fit line
  - c. Implement a multivariate Model using Least Squares and plot best-fit line
  - d. Retrieve model error and model coefficients.
  - e. Observe Variance Inflation Factor(VIF)



Source: <https://www.forbes.com/sites/kalevleertaru/2019/01/15/why-machine-learning-needs-semantic-not-just-statistics/>

**SKILLS:**

- ✓ *Statistical data analysis.*
- ✓ *Classify / Cluster data.*
- ✓ *Tool usage for developing ML applications.*

- f. Implement Ridge regression model
  - g. Implement LASSO regression model
  - h. Report your observations on the above models for house prediction
- Heart disease prediction:
    - a. Implement a logistic regression model to predict whether an individual is suffering from heart disease or not
    - b. Evaluate and compare model performance using the following validation approaches:
      - i. Validation set approach
      - ii. K-fold cross validation
      - iii. Stratified K-fold cross validation
      - iv. LOO strategy
    - c. Plot Confusion matrix
    - d. Report performance of the model in terms of the following metrics:
      - i. Accuracy
      - ii. Precision-Recall
      - iii. F1 Score
    - e. Report your observations and explain when to use what type of measures
  - Implement the Polynomial Regression algorithm to fit data points. Select the appropriatedata set for your experiment and draw graphs.
  - Working with imbalanced datasets:
    - a. Load an imbalanced dataset and visualize imbalance in the data as a bar plot
    - b. Implement KNN model for classification
    - c. Balance the dataset using:
      - i. Random Over sampling
      - ii. Random Under sampling
      - iii. SMOTE
    - d. Implement KNN model for classifying data balanced in the above steps
    - e. Report your observations on the performance of models trained using balanced and imbalanced data
  - Perform effective feature selection in a given dataset using any one of the feature selection techniques.
  - Dimension Reduction:
    - a. Load a dataset and Implement Bayes classification model
    - b. Apply dimension reduction using:
      - i. Principal Component Analysis
      - ii. Linear Discriminant Analysis
    - c. Apply the model on data with reduced dimension
    - d. Compare and contrast model performance in each case

**MODULE-2****UNIT-1****16L+0T+8P=24 Hours****CLASSIFICATION:**

**Classification:** Binary, Multi-class and Multi-label Classification; K-Nearest Neighbours, Support Vector Machines, Decision Trees, The Naïve Bayes' Classifier, Class Imbalance, Perceptron ANN model.

**Ensemble Methods:** Ensemble Learning Model Combination Schemes, Voting, Error-Correcting Output Codes, Bagging: Random Forest Trees, Boosting: Adaboost, Stacking.

**UNIT-2****8L+0T+8P=16 Hours****CLUSTERING:**

**Clustering:** Different distance functions and similarity measures, K-means clustering, Medoids, Hierarchical Clustering-Single linkage and Complete linkage clustering, Graph based Clustering -MST, DBSCAN, Spectral clustering.

**PRACTICES:**

- Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
- Implement the naïve Bayesian classifier for a sample training data set stored as a.csv file. Compute the accuracy of the classifier, considering few test data sets.
- Assuming a set of spam or not-spam mails that need to be classified, use the naïve Bayesian classifier model to perform this task. Calculate the accuracy, precision, and recall for your data set.
- Implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
- Demonstrate the working of the decision tree-based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample?
- Build a model using SVM with different kernels.
- Implement and build models using the following Ensemble techniques
  - a. Bagging
  - b. Boosting: Adaboost, Stacking
- Build a model to perform Clustering using K-means after applying PCA and determining the value of K using the Elbow method.
- Unsupervised Modeling:
  - a. Cluster the data using the following models:
    - i. Spectral Clustering
    - ii. K-medoids
    - iii. DBSCAN
    - iv. Hierarchical Clustering
  - b. Compare and contrast model performance in each case.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                      | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply a wide variety of learning algorithms such as Probabilistic, Discriminative and Generative algorithms for a given application. | Apply        | 1, 2       | 1                |
| 2      | Design an end-to-end Machine-learning model to realize solutions for real-world problems.                                            | Design       | 1          | 3                |
| 3      | Implement various machine learning models using advanced ML tools.                                                                   | Create       | 1, 2       | 5                |
| 4      | Analyze and evaluate the performance of various machine learning models approaches on different kinds of data.                       | Analyze      | 2          | 2                |

**TEXT BOOKS:**

1. Ethem Alpaydin, "Introduction to Machine Learning", 3rd edition, The MIT Press, 2014
2. Flach, Peter. "Machine learning: the art and science of algorithms that make sense of data". Cambridge University Press, 2012.

**REFERENCE BOOKS:**

1. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.
2. Aurélien Géron, "Hands-on Machine Learning with Scikit Learn and Tensor Flow", O'reilly, 2017.
3. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, "An Introduction to Statistical Learning with Applications in R", Springer, 2013. (ISLR).

# 22CS203 OBJECT-ORIENTED PROGRAMMING THROUGH JAVA

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 4 | 4 |

**PREREQUISITE KNOWLEDGE:** Classes, Objects and Inheritance.

## COURSE DESCRIPTION AND OBJECTIVES:

This course is about the fundamentals of Object-Oriented Programming (OOP) Concept and OOP-based software development methodology. Java as a class-based and pure OOP language is used to demonstrate and implement appropriate concepts and techniques. The students are exposed to the concepts, fundamental syntax, and the thought processes behind object-oriented programming. By end of the course, students will acquire the basic knowledge and skills necessary to implement object-oriented programming techniques in software development using Java.

## MODULE-1

### UNIT-1

6L+0T+12P=18 Hours

#### INTRODUCTION:

History of Java, Byte code, JVM, Java buzzwords, OOP principles, Data types, Variables, Scope of variables, Operators, Control statements, Type conversion and casting, Arrays.

Concepts Of Classes and Objects: Introduction to methods, Method over loading, Constructors, Constructor overloading, Usage of static with data and method, Access control, this keyword, Garbage collection, String class, String Tokenizer.

### UNIT-2

10L+0T+20P=30 Hours

#### INHERITANCE AND EXCEPTIONS:

**Inheritance:** Types of inheritance, Member access rules, Usage of super keyword, Method overriding, Usage of final keyword, Abstract classes, Interfaces - differences between abstract classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Packages-defining, creating and accessing a package, importing packages, access control in packages.

**Exception Handling:** Concepts of exception handling, Types of exceptions, Usage of try, catch, throw, throws and finally keywords, Built-in exceptions, User defined exception.

#### PRACTICES:

- There is a telecommunication company called "Powered Air" who have approached you to build their Interactive Voice Response (IVR) system. write a Java program and be able to provide the following menu (given below):

**Note:** User should provide an input for each menu display. Welcome to Powered Air service. What would you like to do?

- |                                         |                          |                    |
|-----------------------------------------|--------------------------|--------------------|
| a. Know my balance.                     | b. Know my validity date |                    |
| c. Know number of free calls available. | d. More                  |                    |
| 1. Prepaid Bill Request                 | 2. Customer Preferences  | 3. GPRS activation |
| 4. Special Message Offers               | 5. Special GPRS Offers   | 6. 3G Activation   |
| 7. Go back to Previous menu             |                          |                    |

You are free to display your own messages in this IVR.

- Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.

Source: <https://www.datasciencecentral.com/wp-content/uploads/2021/10/8667507462.jpeg>

**Hint:** Area of rectangle = length \* width, Perimeter of rectangle = 2\*(length+width).

- Implement a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (use StringTokenizer class).
- Implement a java program to print all tokens of a string on the bases of multiple separators (use StringTokenizer class).
- Using inheritance, one class can acquire the properties of others. Consider a class Animal that has only one method “walk”. Next, create a Bird class that also has a fly method. Finally, create a bird object that can both fly and walk.
- Using inheritance, Write the following code in your editor :
  1. A class named Arithmetic with a method named “add” that takes integers as parameters and returns an integer denoting their sum.
  2. A class named Adder that inherits from a superclass named Arithmetic.

**Note:** Your classes should not be Public.

- When a subclass inherits from a superclass, it also inherits its methods; however, it can also override the superclass methods (as well as declare and implement new ones). Consider the Sports class having methods getName()[which returns name of sport] and getNumberOfTeamMembers()[which returns noof team members] create a Soccer class that inherits from the Sports class. We can override the get Name method and return a different subclass-specific string and override getNumberOfTeamMembers method and return noof team members
- Implement a java program to create an abstract class named Shape that contains an empty method named number Of Sides ( ). Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
- You are given an interface Advanced Arithmetic which contains a method signature int divisor\_sum(int n). You need to write a class called My Calculator which implements the interface. divisor\_sum function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so divisor\_sum should return 12. The value of n will be at most 1000.
- Implement a Java program for the following
  - a) Creation of simple package.
  - b) Accessing a package.
- Implement a Java program to read two numbers a,b from user and perform division a/b,if the user passes b value as zero, handle the exception using try and catch otherwise display the result.
- Create a class called Customer with data members account\_number, balance (initialize with 10000), and member functions print(), deposit(), and withdraw(). Print method display account number and balance. If withdraw amount is less than current balance while withdrawing, throw an exception “In Sufficient Funds”. If the input is 1 do print. If the input is 2 withdraw (). If the input is 3 deposit. If the input is 4 terminate program.
- Implement a Java program which accepts age as input from the user and throws an exception “Not Eligible to Vote” when age is <=18 otherwise print “Eligible to Vote”.

## MODULE-2

### UNIT-1

8L+0T+16P=24 Hours

#### MULTI THREDDING AND FRAMEWORK:

**Multithreading:** Concepts of multi threading, Differences between process and thread, Thread life cycle, Creating multiple threads using Thread class and Runnable interface, Synchronization, Thread priorities, Inter thread communication.

**Collection Framework:** Collections Overview, Collection Interfaces - List, Set, Map, List – Array List, Linked List, Vector, Set - HashSet, TreeSet, Map - HashTable, HashMap, accessing a collection via an Iterator, comparator, comparable.

#### SKILLS:

- ✓ To analyse and develop algorithm for real life problems using Java.
- ✓ Experience with developing and debugging programs in different IDEs.
- ✓ Develop multi-threaded applications.
- ✓ Creating web applications

## UNIT-2

8L+0T+16P=24 Hours

**SWINGS:**

**GUI Programming With Swing:** Delegation event model-Events, Event sources, Event Listeners, Event classes, handling mouse and keyboard events.

**Exploring Swing Controls:** JLabel and Image Icon, JTextField, JButton, JCheckBox, JRadioButton, JTabbed Pane, JList, JCombo Box.

**PRACTICES:**

- Print in Order

Suppose we have a class:

```
public class Foo {
 public void first() { print("first"); }
 public void second() { print("second"); }
 public void third() { print("third"); }
}
```

The same instance of Foo will be passed to three different threads. Thread A will call first(), thread B will call second(), and thread C will call third(). Design a mechanism and modify the program to ensure that second() is executed after first(), and third() is executed after second().

**Note:** We do not know how the threads will be scheduled in the operating system, even though the numbers in the input seem to imply the ordering. The input format you see is mainly to ensure our tests' comprehensiveness.

**Example 1:**

Input: nums = [1,2,3]      Output: "firstsecondthird"

**Explanation:** There are three threads being fired asynchronously. The input [1,2,3] means thread A calls first(), thread B calls second(), and thread C calls third(). "firstsecondthird" is the correct output.

**Example 2:**

Input: nums = [1,3,2]      Output: "firstsecondthird"

**Explanation:** The input [1,3,2] means thread A calls first(), thread B calls third(), and thread C calls second(). "firstsecondthird" is the correct output.

- **Flood Fill:**

An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image.

You are also given three integers sr, sc, and color. You should perform a flood fill on the image starting from the pixel image[sr][sc].

To perform a flood fill, consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color), and so on. Replace the color of all of the aforementioned pixels with color.

Return the modified image after performing the flood fill.

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 0 |
| 1 | 0 | 1 |

→

|   |   |   |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 2 | 0 |
| 2 | 0 | 1 |

**Example 1:**

**Input:** image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2

**Output:** [[2,2,2],[2,2,0],[2,0,1]]

**Explanation:** From the centre of the image with position (sr, sc) = (1, 1) (i.e., the red pixel), all pixels connected by a path of the same color as the starting pixel (i.e., the blue pixels) are colored with the new color.

Note the bottom corner is not coloured 2, because it is not 4-directionally connected to the starting pixel.

**Example 2:**

**Input:** image = [[0,0,0],[0,0,0]], sr = 0, sc = 0, color = 0

**Output:** [[0,0,0],[0,0,0]]

**Explanation:** The starting pixel is already colored 0, so no changes are made to the image.

- **Count words in a given string**

The input parameter is a list of strings representing lines of text.

Count how often the word occurs in the text.

If the word “kitten” occurred in a text 23 times, then its entry would be “kitten - 23\n”. Return statistics as a String containing all the entries.

Omit all words which contain less than 4 letters and appear less than 10 (the words which are too small or too rare) The entries in the resulting String should be also sorted by their amount and then in alphabetical order if it is needed.

- Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
- Implement a Java program for handling key events when the key board is pressed, released, typed.
- Implement a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button “add” is pressed.
- Implement a Java program to design student registration form using Swing Controls. The form which having the following fields and button “save”. Form Fields are: Name, RNO, Mail id, Gender, Branch, and Address.
- Implement a java program using swings to design a multiple choice question having three options (use radio button) ,display the message using dialog box “Your answer is wrong” if the user selects wrong option otherwise display ,”Your answer is correct.”

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                  | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply object oriented concepts on real time scenarios.                                                           | Apply        | 1          | 1,2              |
| 2      | Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes            | Apply        | 1, 2       | 1,2              |
| 3      | Design and develop Java applications to solve real world problems by using modern tools and collection framework | Create       | 2          | 3,5              |
| 4      | Design and develop GUI based applications using swings for internet and system based applications.               | Create       | 2          | 3,5              |

**TEXT BOOKS:**

1. Herbert Schildt, “Java the complete reference”, 12th Edition, McGraw Hill, Education, 2021.
2. M.T. Somashekara, D. S. Guru, K.S. Manjunatha, “Object-Oriented Programming with Java”, 1st Edition, PHI Learning, 2017.

**REFERENCE BOOKS:**

1. E. Balagurusamy, “Programming with Java”, 6th Edition, McGraw Hill, 2019.
2. Mark Lassoﬀ, “ Java Programming for Beginners: Learn the fundamentals of programming with Java”, 1st Edition, Packt Publishing Limited,2017.
3. Philip Conrod, Lou Tylee, “ Learn Java GUI Applications : A JFC Swing Tutorial”, 11th Edition, Kidware Software, 2019.



# 22CS207 OPERATING SYSTEMS



Source: [https://www.123rf.com/stock-photo/operating\\_system.html](https://www.123rf.com/stock-photo/operating_system.html)

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 2 | 3 |

**PREREQUISITE KNOWLEDGE:** Knowledge of computers fundamentals, Computer organization & Digital logic and its design.

## COURSE DESCRIPTION AND OBJECTIVES:

This course aims at concepts and principles of Operating Systems, its overall responsibility in acting as an interface between the system's hardware components and the user. Further, it also helps students to understand the different scheduling policies, process synchronization mechanisms, deadlock handling mechanisms and memory management techniques.

## MODULE-1

### UNIT-1

**10L+0T+10P=20 Hours**

#### LINUX FILE SYSTEM & PROCESS SCHEDULING:

**Introduction to LINUX File System:** The LINUX file System, File System Hierarchy, File system Commands, File Attributes, File Permissions.

**Filters:** cmp, comm, diff, head, tail, find, cut, paste, sort, uniq.

**Regular Expressions:** grep, egrep, fgrep, Sed- line addressing, context addressing, text editing, substitution.

**Introduction to Operating System:** What Operating System do; Operating System Structure; Process concept-overview, Process Scheduling, Operations on Process; Inter Process Communication; Threads; Process (CPU) Scheduling-Scheduling Criteria, Scheduling Algorithms; Multiple-Processor scheduling;

### UNIT-2

**6L+0T+6P=12 Hours**

#### PROCESS SYNCHRONIZATION AND DEADLOCKS:

**Process Synchronization:** The critical-section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

**Deadlocks:** Deadlock characterization; Methods of handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery.

#### PRACTICES:

- Use the cat command to create a file containing the following data. Call it mytable.txt use tab to separate the fields.
 

|      |      |       |
|------|------|-------|
| 1425 | ravi | 15.65 |
| 4320 | ramu | 26.27 |
| 6830 | sita | 36.15 |
| 1450 | raju | 21.86 |

  - a. Use the cat command to display the file, mytable.txt.
  - b. Use the vicomm and to correct any errors in the file, mytable.txt.
  - c. Use the sort command to sort the file mytable.txt according to the first field.
  - d. Call thesortedfilemytable.txt (same name)
  - e. Printthefilemytable.txt.
  - f. Use the cut & paste commands to swap fields 2 and 3 mytable. Call it mytable.txt (same name)
  - g. Print the new file, mytable.txt.



- Write a shell script that takes a command-line argument and reports on whether it is directory, a file, or something else.
- Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.
- Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
- Write a shell script that computes the total and average marks of a student according to the following;
  - If average marks  $\geq 69$  then result is -Distinction.
  - If average marks  $\geq 59$  and  $\leq 70$  then result is -First Class.
  - If average marks  $\geq 49$  and  $\leq 60$  then result is -Second Class. If average marks  $\leq 50$  then result is -Pass.
  - Note that any subject marks  $\leq 40$  then result is -Fail.
  - Accept student name and six subject marks through the keyboard.
- Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
- Write a shell script, which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.
- Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
- Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
- Implementation of new process creation and its communications.
- Implement of thread creation and deletion.
- Implementation of FCFS scheduling.
- Implementation of SJF and RR Scheduling.
- Implementation of producer consumer problem.
- Implementation of Banker's algorithm for Dead lock avoidance.

## MODULE-2

### UNIT-1

8L+0T+8P=16 Hours

#### MEMORY MANAGEMENT:

**Memory Management:** Basic concept of memory management, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

**Virtual Memory Management:** Demand Paging, Page Replacement: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU), Least Recently used (LRU), Allocation of Frames.

### UNIT-2

8L+0T+8P=16 Hours

#### SECONDARY STORAGE STRUCTURE:

**Secondary Storage Structure:** Over view of mass-storage structure, disk structure, disk scheduling;

**File System Interface** - File concept, Access Methods, Directory & Disk Structure, File-System Mounting, File Sharing, Protection; File-system structure.

**File System Implementation**- Directory implementation, Allocation Methods, Free Space Management.

#### PRACTICES:

- Assume that you have a page-reference string for a process with  $m$  frames (initially all empty). The page-reference string has length  $p$ , and  $n$  distinct page numbers occur in it.
  - a) What is a lower bound on the number of page faults?
  - b) What is an upper bound on the number of page faults?
- Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from "bad" to "perfect" according to their page-fault rate. Separate those algorithms that suffer from Belady's anomaly from those that do not.

#### SKILLS:

- ✓ Manage open-source operating systems like Ubuntu, Fedora etc.
- ✓ Understand the concepts of Processes scheduling and File Systems.
- ✓ Identification of different disk scheduling methodologies.

- a) LRU replacement.      b) FIFO replacement.  
 c) Optimal replacement.    d) Second-chance replacement.
- Consider the page reference string: 1, 2, 3, 4,2, 1, 5, 6, 2, 1, 2, 3,7, 6, 3, 2, 1, 2, 3, 6.
  - How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.
    - LRU replacement
    - FIFO replacement
    - Optimal replacement
  - How many page fault occur for your algorithm for the following reference string with four page frames? 1, 2, 3,4, 5, 3, 4, 1, 6,7, 8, 7, 8, 9, 7, 8,9, 5, 4, 5,4, 2.
  - What is the minimum number of page faults for an optimal page replacement strategy for thereference string above with four page frames?
  - Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. Three alternative results are shown below. For each case, what is happening?
    - a) Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?
    - b) CPU utilization 13 percent; disk utilization 97 percent.
    - c) CPU utilization 87percent; disk utilization 3 percent.
    - d) CPU utilization 13 percent; disk utilization 3 percent.
  - Implementation of Disk scheduling algorithm–FCFS.
  - Implementation of Disk scheduling algorithm–SSTF and SCAN.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                              | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Classify the basic concepts of operating system and explore Linux ecosystem.                                                                 | Analyze      | 1          | 1                |
| 2      | Apply the concepts of process scheduling algorithms and process synchronization techniques to derive the efficiency of resource utilization. | Apply        | 1          | 1, 2, 3, 5, 12   |
| 3      | Analyze the requirements for attempting Operating systems principles.                                                                        | Analyze      | 1, 2       | 1, 2, 12         |
| 4      | Design the various memory management schemes For a given scenario.                                                                           | Create       | 2          | 3, 5             |
| 5      | Apply the concepts of file system interface and implementation.                                                                              | Apply        | 1, 2       | 2, 5             |

**TEXT BOOKS:**

1. Sumitabha Das, Unix concepts and applicationsII, TMH Publications, 4th Edition, July 2017.
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 9th Edition, John Wiley & SonsInc, 2013.

**REFERENCE BOOKS:**

1. Richard. Stevens and Stephen A Rago, "Advanced Programming in the Unix Environment", 3rd Edition, Addison-Wesley, 2013.
2. William Stallings, "Operating Systems-Internals and Design principles" PHI, 7th Edition, 2012.
3. Gary J. Nutt. Addison-Wesley, "Operating Systems: A Modern Perspective", 2nd Edition, 2001.
4. B.A. Forouzan & R.F.Giberg, - Unix and shell ProgrammingII, Thomson, 1st Edition, New Delhi, 2003.

**III**  
YEAR

**B.Tech.**

**COURSE CONTENTS**

I SEM & II SEM

# CSE - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

## I SEMESTER

|   |         |                                                                      |
|---|---------|----------------------------------------------------------------------|
| ▶ | 22TP301 | - Soft Skills Laboratory                                             |
| ▶ | 22AM301 | - Deep Learning                                                      |
| ▶ | 22CS204 | - Computer Networks                                                  |
| ▶ | 22DS203 | - Formal Languages and Automata Theory                               |
| ▶ | 22AM302 | - Inter-Disciplinary Project - Phase - I                             |
| ▶ | 22AM303 | - Industry Interface Course                                          |
| ▶ |         | - Department Elective - 1                                            |
| ▶ |         | - NCC/NSS/SAC/E-Cell/Student Mentoring/Social Activities/Publication |

## II SEMESTER

|   |         |                                               |
|---|---------|-----------------------------------------------|
| ▶ | 22TP302 | - Quantitative Aptitude and Logical Reasoning |
| ▶ | 22AM304 | - Fundamentals of Image Processing            |
| ▶ | 22AM305 | - Reinforcement Learning                      |
| ▶ | 22CS303 | - Web Technologies                            |
| ▶ | 22AM306 | - Inter-Disciplinary Project - Phase - II     |
| ▶ |         | - Department Elective - 2                     |



# 22TP301 SOFT SKILLS LABORATORY

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 2 | 1 |

**PREREQUISITE KNOWLEDGE:** Grasp on their own academic achievements.

## COURSE DESCRIPTION AND OBJECTIVES:

To impart employability skills like resume preparation and facing interviews. To enable trainees to develop interpersonal and leadership skills and to train them on work place skills like making presentations, participating in group discussions etc.

## MODULE-1

### UNIT-1

**0L+0T+8P=8 Hours**

#### PERSONALITY DEVELOPMENT:

**Soft Skills:** Need for soft skills, professionalism, employability skills; **Communication:** Need for effective communication - the process of communication, levels of communication, flow of communication, choice of diction and style with reference to setting (formal, semi-formal or informal); communication networks, barriers to communication, miscommunication, noise and ways to overcome the barriers; **Career Planning:** Job vs. career, SWOT analysis.

### UNIT-2

**0L+0T+8P=8 Hours**

#### LANGUAGE AND VOCABULARY:

**Vocabulary Building:** Word etymology, roots, prefixes & suffixes, synonyms & antonyms, collocations, one-word substitutes, analogies, idioms and phrases, contextual guessing of unfamiliar words, task-oriented learning; Reflection of language on Personality, Gender sensitive language in MNCs, Mind your language, Seven essential skills for a team player; attentive listening, intelligent questioning, gently persuading, respecting other's views, assisting others, sharing, participating actively.

#### PRACTICES:

- Self-Introduction.
- Personal and Academic SWOC.
- Johari Window.
- Giving and taking opinions of Self Vs others and assessing oneself.
- Goal setting.
- Short, Mid and Long Term goals planning the semester.
- Time management: four quadrant system.
- Stephen Covey Time Management Matrix planning a semester.
- Stress-management.
- Questionnaire to assess level of stress.
- 50 words towards resume preparation and interviews.
- Newly coined words.
- Gender sensitive words and Words acceptable in Indian context and objectionable international context.

## MODULE-2

### UNIT-1

**0L+0T+8P=8 Hours**

#### LANGUAGE IN ACTION:

**Functional English:** Situational dialogues, Role plays (including small talk); **Group Discussion:** Articulation and flow of oral presentation, dynamics of group discussion, intervention, summarizing and conclusion, voice modulation, content generation, Key Word Approach (KWA), Social, Political, Economic,



Source: <https://choosework.ssa.gov/blog/2019-07-23-soft-skills-an-intro-to-effective-communication>

**SKILLS:**

- ✓ *Balance social and emotional intelligence quotients through SWOC, JOHARI etc. activities.*
- ✓ *Prepare tailor made resume and face various job interviews with enriched personality traits.*
- ✓ *Career planning with clear personal and professional goals.*
- ✓ *Solve personal and professional life hiccups with confidence and maturity*

Legal and Technical Approach (SPELT), View Point of Affected Part (VAP), language relevance, fluency and coherence – 11th and 12th weeks; Resume preparation: Structure and presentation, defining career objective, projecting one's strengths and skill-sets, summarizing, formats and styles and covering letter-Statement of Purpose.

**UNIT-2****0L+0T+8P=8 Hours****PREPARING FOR PRESENTATIONS AND INTERVIEWS:**

**Facing Interviews:** Interview process, understanding employer expectations, pre-interview planning, opening strategies, impressive self-introduction, answering strategies, other critical aspects such as body language, grooming, other types of interviews such as stress-based interviews, tele- interviews, video interviews, frequently asked questions (FAQs) including behavioral and HR questions and the aspect looked at by corporate during interviews; **Presentation Skills:** Selection of a topic, preparing an abstract, gathering information, organizing the information, drafting the paper, citing reference sources – writing striking introductions, discussing the methodology used, developing the argument, presentation style, language, presenting the paper and spontaneously answering audience questions.

**PRACTICES:**

- Opening and closing a telephonic conversation.
- Making an appointment.
- Making a query.
- Offering/Passing on information.
- Communicating with superiors.
- Expressing agreement/objection.
- Opening bank account (combination of prepared and impromptu situations given to each student).
- Group Discussions on various topics.
- Preparing SoP and Resume.
- Mock interviews on the FAQs including feedback.
- Oral presentation with the help of technology (Preparing PPT and presenting).

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                          | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Have the ability to introspect on individual strengths and weaknesses, and emerge as a balanced personality with improved self-awareness and self-worth. | Apply        | 1          | 12               |
| 2      | Observe gender sensitive language and workplace etiquette in his professional life.                                                                      | Analyze      | 1          | 9                |
| 3      | Be able to prepare a resume and gain the confidence to face an interview.                                                                                | Create       | 1, 2       | 10               |
| 4      | Possess the interpersonal skills to conduct himself/herself effectively in everyday professional and social contexts.                                    | Apply        | 2          | 8                |
| 5      | Bring professionalism into his/her daily activities.                                                                                                     | Create       | 2          | 8                |

**TEXT BOOKS:**

1. Adrian Furnham, "Personality and intelligence at work", Psychology Press, 2008.
2. S. P. Dhanvel, "English and Soft skills", Orient Blackswan, 2011.

**REFERENCE BOOKS:**

1. Edward Holffman, "Ace the corporate personality", McGraw Hill, 2001.
2. John Adair Kegan Page, "Leadership for innovation", Kogan, 2007.
3. Krishna Mohan & NP Singh, "Speaking English effectively", Macmillan, 2008.
4. Rajiv K. Mishra, "Personality Development", Rupa & Co. 2004.

**22AM301 DEEP LEARNING**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Machine Learning, Python Programming.**COURSE DESCRIPTION AND OBJECTIVES:**

This course offers sufficient details required to understand the basic building blocks of various deep learning based models. Especially, focuses on different types of neural network models like feed forward neural networks, convolutional neural networks, recurrent neural networks, and deep auto encoders. During this course the students build, train, and evaluate deep neural network models for various applications in image, text, and speech domains. In addition, throughout this course students will be able to understand hyper parameter tuning and other best-practices to be followed while training deep neural network models.

**MODULE-1****UNIT-1****12L+0T+8P=20 Hours****EVOLUTION OF DEEP NEURAL NETWORKS:**

**Deep Learning Intuition:** History of Deep Learning, What is Deep Learning? Applications of Deep Learning.

**Neural Network Basics:** McCulloch–Pitts neuron, Perceptron learning rule, Perceptron convergence theorem, Sigmoidal neuron, Multi-layer feed forward neural network, back propagation algorithm, Gradient descent method, Stochastic gradient descent method. Shallow Neural Networks and Deep Neural Networks

**REGULARIZATION and OPTIMIZATION for training Deep Models:** Optimization methods - Adagrad, Adadelta, RMSProp, Adam; Regularization Methods-Dropout, Drop connect, Batch normalization; Activation functions - Linear, sigmoid, sigmoid, ReLU and variations of ReLU; Loss Function, Improving the training process – Dataset Augmentation, Noise Robustness, Weight Initialization methods, Early stopping, Parameter sharing and tying, bagging and other ensemble methods;

**UNIT-2****12L+0T+8P=20 Hours****CONVOLUTIONAL NEURAL NETWORKS:**

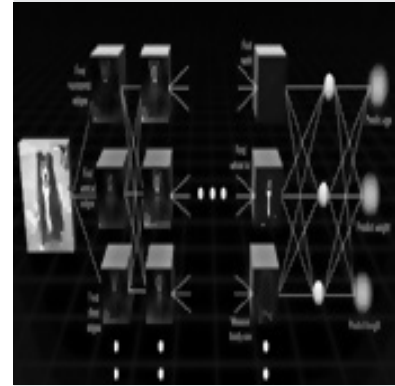
Foundations of Convolutional Neural Network, Popular Deep CNN Models: LeNet, AlexNet, VGGNet, ResNet, Google Net and other architectures.

**Instructions for Practices:**

- Practice Assignments can be implemented using the Keras / Tensorflow APIs of Python
- Relevant data sets can be downloaded from standard repositories such as Kaggle/UCI or can be developed by the students.

**PRACTICES:**

- Implement Logistic regression With Neural Network Mindset
  - logistic regression classifier for classification
  - Plot the loss over each epoch
  - Plot the accuracy over each epoch
  - Report final Accuracy
- Implement Shallow Neural Network model:



Source: [https://www.symmetrymagazine.org/sites/default/files/images/standard/neural\\_network\\_visual\\_final.jpg](https://www.symmetrymagazine.org/sites/default/files/images/standard/neural_network_visual_final.jpg)

**SKILLS:**

- ✓ *Developing Vision and text based applications*
- ✓ *Hyperparameter Tuning of a deep Neural network model.*
- ✓ *Tensor Flow/ Keras tool usage for neural network implementation.*

- Implement a binary classification neural network with a single and multiple hidden layers
- Implement a Multi-class classification neural network with a single and multiple hidden layers
- Vary the number of neurons at suitable layers
- Hyper parameter Tuning of a Neural Network model implemented for hand-written digit classification:
  - Vary the type of activation functions
  - Choose suitable Loss functions
  - Vary the number of neurons at suitable layers
  - Vary Weight Initialization methods
  - Save the Best Model and load the saved model
- Building a Deep Neural Network:
  - Implement a multi-class classification neural network with number of layers of your choice
  - Include Batch Normalization layers
  - Vary Optimization methods
  - Add drop out layers
- Convolutional Neural Network Models
  - Design a Convolutional neural network with the layers of your choice
  - Compare the performance by changing the
    - Kernel size
    - Number of feature maps at each convolutional layer
    - Stride
    - Padding
    - Number of fully connected layers
- Visualization of CNN Models
  - Design a Convolutional Neural Network Model for image classification
  - Plot Model Architecture
  - Visualize feature maps after training of CNN
  - Visualize class activation maps

**MODULE-2****UNIT-1****14L+0T+10P=24 hours****DEEP UNSUPERVISED LEARNING:**

**Transfer learning Approaches:** Deep Pre-trained architectures- AlexNet, VGG16, VGG19, ResNet. Use deep Convolutional architectures for feature extraction and fine-tuning tasks.

**Deep Unsupervised Learning:** Autoencoders- Under complete Autoencoders, regularized auto encoders, Representation power, layer size and depth, stochastic encoders and decoders, Denoising auto-encoders, Sparse auto encoder, Contractive auto-encoders.

**UNIT-2****10L+0T+6P=16 hours****RECURRENT NEURAL NETWORKS:**

Architecture of an RNN, unfolding of an RNN, Backpropagation through time, Long short term memory (LSTM), Gated recurrent units, Applications- Text Classification, Sentiment Analysis.

**PRACTICES:**

- Using Deep pre-trained CNN model for feature extraction:
  - Extract features from the FC1 of VGG network
  - Train any traditional ML model like SVM for classification
  - Repeat the above by considering FC2 of VGG for feature extraction
- Fine-tuning Deep pre-trained CNN for Classification:
  - Fine-tune VGG network for the task under consideration



- Check the performance by making
  - all the layers trainable
  - freezing the initial layers
  - freezing the entire network except the final layer
- Design MLFFNN with 3-level stacked autoencoder based pre-training for Black and white image data, Display features extracted by different levels of stacked autoencoder at the end of pre-training.
- Sentiment Analysis
  - Pre-process the text
  - Convert the text into word embeddings
  - Implement the classification network using LSTMs/ GRUs
  - Pre-process the text
  - Convert the text into word embeddings
  - Implement the classification network using LSTMs/ GRUs

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                   | Blooms Level | Module No. | Mapping with POs |
|--------|-----------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Implementation of Deep learning models to solve various real-time problems.       | Apply        | 1, 2       | 1, 3             |
| 2      | Analyze performance of a deep network and tune its capacity and hyper parameters. | Analyze      | 1          | 2                |
| 3      | Leveraging tools to Build deep networks and apply them for real word tasks.       | Apply        | 1, 2       | 1, 5             |
| 4      | Developing core components for deep learning algorithms.                          | Design       | 1, 2       | 3                |

**TEXT BOOKS:**

1. Ian Goodfellow and Yoshua Bengio and Aaron, "Deep Learning", 1st Edition, An MIT Press Book, 2016.
2. Charu C. Aggarwal "Neural Networks and Deep learning" Springer International Publishing, 2018.

**REFERENCE BOOKS:**

1. Francois Chollet, "Deep learning with python", 1st edition, Manning Publications, 2017.
2. S. Haykin, "Neural Networks and Learning Machines", 3rd edition, Prentice Hall of India, 2011.
3. Josh Patterson and Adam Gibson, "Deep Learning: A Practitioner's Approach", 1st Edition, O'Reilly, 2017.
4. Satish Kumar, "Neural Networks, A Classroom Approach", Tata McGraw -Hill, 2007.

# 22CS204 COMPUTER NETWORKS



Source: <https://snabaynetworking.com/what-is-computer-network-and-its-types/>

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** JAVA programming and UNIX commands.

## COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on imparting knowledge about various protocols involved in LANs and WANs. In addition, it gives a good foundation on different protocols such as data link protocols, internet protocols, and transport protocols present in the respective layers of the data communication system.

## MODULE-1

### UNIT-1

**8L+0T+6P=14 Hours**

#### INTRODUCTION TO COMPUTER NETWORKS AND INTERNET:

Understanding of network and Internet, the network edge, the network core, Understanding of Delay, Loss and Throughput in the packet switching network, protocols layers and their service model, History of the computer network.

### UNIT-2

**16L+0T+10P=26 Hours**

#### APPLICATION LAYER & TRANSPORT LAYER:

Principles of computer applications, Web and HTTP, E-mail, DNS, Socket programming with TCP and UDP.

Introduction and transport layer services, Multiplexing and Demultiplexing, Connectionless transport (UDP), Principles of reliable data transfer, Connection-oriented transport (TCP), Congestion control.

#### PRACTICES:

- Install Network Simulator 2/3. Create a wired network using dumbbell topology. Attach agents, generate both FTP and CBR traffic, and transmit the traffic. Vary the data rates and evaluate the performance using metric throughput, delay, jitter and packet loss.
- Create a static wireless network. Attach agents, generate both FTP and CBR traffic, and transmit the traffic. Vary the data rates and evaluate the performance using metric throughput, delay, jitter and packet loss.
- Create a mobile wireless network. Attach agents, generate both FTP and CBR traffic, and transmit the traffic. Vary the data rates and evaluate the performance using metric throughput, delay, jitter and packet loss.
- Implementation of one-way and two-way communication using TCP / UDP.
- Hello command is used to know whether the machine at the other end is working or not. Echo command is used to measure the round trip time to the neighbour. Implement Hello and Echo commands using JAVA.

## MODULE-2

### UNIT-1

**12L+0T+8P=20 Hours**

#### NETWORK LAYER:

Introduction to forwarding and routing, Network Service models, Virtual and Datagram networks, study of router, IP protocol and addressing in the Internet, Routing algorithms, Broadcast and Multicast routing.

**UNIT-2****12L+0T+8P=20 Hours****THE LINK LAYER AND LOCAL AREA NETWORKS:**

Introduction to link layer services, error detection, and correction techniques, Multiple access protocols, addressing, Ethernet, switches, and VLANs.

**PRACTICES:**

- Find all the IP addresses on your network using Unicast, Multicast, and Broadcast on your network.
- Use Packet tracer software to build network topology and configure using Distance vector routing and Link State routing protocols.
- Study different types of Network cables (Copper and Fiber) and prepare cables (Straight and Cross) to connect two or more systems.
  - a. Use a crimping tool to connect jacks.
  - b. Use a LAN tester to connect the cables.
  - c. Install and configure Network Devices: HUB, Switch and Routers (Consider both manageable and non-manageable switches. Perform logical configuration of the system and set the bandwidth of different ports).
  - d. Install and Configure Wired and Wireless NIC and transfer files between systems in Wired LAN and Wireless LAN. Consider both ad-hoc and infrastructure modes of operation.
- Apply the commands such as Ping, Tracert, Ipconfig, pathping, telnet, FTP, getmac, ARP, Hostname, Nbtstat, netdiag, and Nslookup to solve various problems.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                      | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Build the basic concepts of Network hardware, software and reference models.         | Apply        | 1          | 1, 2, 12         |
| 2      | Evaluate different physical layer media and switching methods.                       | Evaluation   | 1          | 1, 2, 5, 12      |
| 3      | Implement various protocols with modern tools.                                       | Apply        | 1          | 1, 2, 3, 5, 12   |
| 4      | Apply different protocols to perform end-to-end delivery and interaction with users. | Analyze      | 2          | 1, 2, 12         |
| 5      | Analyze various design issues, protocols and functionalities of network layer.       | Analyze      | 2          | 1, 2, 12         |
| 6      | Demonstrate various protocols involved in data link layer operations.                | Apply        | 2          | 1,2, 5           |

**TEXT BOOKS:**

1. Kurose and Ross, "Computer Networking- A Top-Down approach", 6th Edition, Pearson, 2017.
2. Behrouz Forouzan, "Computer Networks- A Top-Down approach", McGraw Hill, 2014.

**REFERENCE BOOKS:**

1. Andrew S. Tanenbaum, "Computer Networks", 5th Edition. Pearson Education, 2014.
2. Behrouz A. Forouzan, "Data communications and Networking", 5th Edition, TMH, 2017.
3. William Stallings, "Data and Computer Communications", 10th Edition, Pearson Education, 2017.
4. Fred Halsall, "Computer Networking and the Internet", 5th edition, Addison Wesley, 2005.

**SKILLS:**

- ✓ Establish local area networks with different topologies.
- ✓ Design of new routing protocols.
- ✓ Network troubleshooting such as installing network interface card drivers, setting IP addresses, subnet masking, etc.

# 22DS203 FORMAL LANGUAGES AND AUTOMATA THEORY

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 0 | 3 |

**PREREQUISITE KNOWLEDGE:** Knowledge of graphs, trees and logic.

## COURSE DESCRIPTION AND OBJECTIVES:

This course aims to teach the student to identify different formal language classes and their relationships, strong theoretical foundation for designing compilers. In addition to this the student will be able to learn the techniques for information processing, design different grammars, automata and recognizers for different formal languages.

## MODULE-1

### UNIT-1

8L+8T+0P=16 Hours

#### INTRODUCTION:

Alphabets, Strings and languages, Automata and Grammars, Regular languages, Chomsky hierarchy of languages, Deterministic finite automata (DFA)-Formal definition, Simplified notation, State transition graph, Transition table, Language of DFA; Nondeterministic finite automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of finite automata, FA with output - Moore and Mealy machine, Equivalence of Moore and Mealy machine, Applications and Limitation of FA.

### UNIT-2

8L+8T+0P=16 Hours

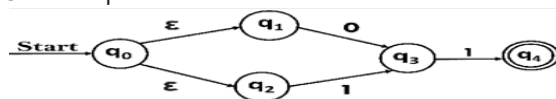
#### REGULAR EXPRESSION:

**Regular Expression (RE):** Definition, Operators of regular expression and their precedence, Algebraic laws for Regular Expressions, Kleen's Theorem, Regular Expression to FA, DFA to regular expression, Arden theorem, Non regular languages, pumping lemma for regular languages (proofs not Required), Application of pumping lemma, Closure properties of regular languages, Decision properties of regular languages.

**Grammar Formalism:** Regular Grammars-Right linear and left linear grammars, Equivalence between regular linear grammar and FA;

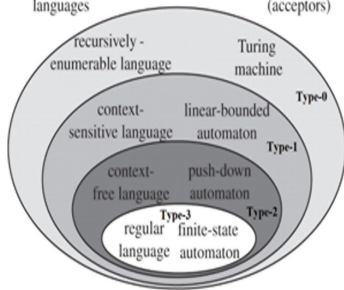
#### PRACTICES:

- Design DFA and NFA which accepts the following languages over the alphabet  $\{0,1\}$ . And also convert NFA to DFA. Give separate Automata for each and also write RE for the obtained automata.
  - The set of all strings ends with 00,
  - with three consecutive 0's
  - with 011 as a substring
  - either begin or ends with 01
  - strings whose fourth symbol from the right end is 1
  - even number of 0's
  - number of 1's are divisible by three
- Design NFA to recognize the following set of strings
  - abc, abd,andaacd: Assume the alphabet is  $\{a,b,c,d\}$
  - 0101,101 and 011: Assume the alphabet is  $\{0,1\}$
  - ab,bc and ca: Assume the alphabet is  $\{a,b,c\}$
- Convert epsilon NFA to DFA.



grammars (generators) and languages

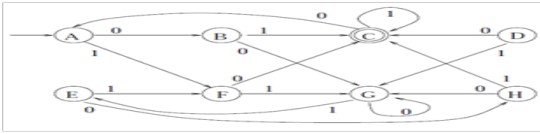
automata (acceptors)



Chomsky hierarchy

Source: <https://sameer9247.wordpress.com/2016/11/15/theory-of-computation/>

- Minimize the following DFA.



- Construct Mealy and Moore Machines and equivalent them for the residue (remainder) mod 3 of binary input.
- Construct Finite Automata for the following Regular Expressions
  - $RE=ab(a+b)^*$
  - $RE=(a+ab)(ab+ab)^*$
- Prove that the following languages are not Regular
  - $L=\{ap \mid p \text{ is a prime number}\}$
  - $L=\{bn \mid n=i^2 \text{ and } i>1\}$
  - $L=\{WWR \mid W \text{ is } (a,b)^*\}$
  - $L=\{anbn+1 \mid n \geq 1\}$

## MODULE-2

### UNIT-1

8L+8T+0P=16 hours

#### CONTEXT FREE GRAMMAR:

Definition, Examples, Derivation, Derivation trees, Ambiguity in grammar, Inherent ambiguity, Ambiguous to unambiguous CFG, Useless symbols, Simplification of CFGs; Normal forms for CFGs - CNF and GNF, CFLs; Closure properties of Decision properties of CFLs-Emptiness, Finiteness and membership, pumping lemma for CFLs (proofs not Required), Application of pumping lemma.

### UNIT-2

8L+8T+0P=16 hours

#### PUSH DOWN AUTOMATA (PDA):

**Push Down Automata (PDA):** Description and definition, Instantaneous description, Language of PDA, Acceptance by final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

**Turing Machines (TM):** Basic model, Definition and representation, Instantaneous Description, Language acceptance by TM, Computable functions, Types of Turing Machines, Universal TM, Recursive and Recursively Enumerable Languages, undecidability.

#### PRACTICES:

- Construct CFG for the following
  - $L=\{anbn \mid n>1\}$
  - $L=\{WWR \mid W \text{ is } (a,b)^*\}$
  - $L=\{ap \mid p \text{ is a prime}\}$
- Derive the strings 10001 using left most derivation and right most derivation and parse tree by using the following grammar. And show that grammar is ambiguous
 
$$S \rightarrow T000T$$

$$T \rightarrow 0T \mid 1T$$
- Convert the following CFG to CNF
 
$$S \rightarrow ABC \mid Aa$$

$$A \rightarrow a$$

$$B \rightarrow d \mid$$

$$C \rightarrow Aab \mid a$$
- Convert the following CFG to GNF
 
$$S \rightarrow AA \mid 0$$

$$A \rightarrow SS \mid 1$$
- Prove that the following are not CFL
  - $L=\{ap \mid p \text{ is a prime number}\}$
  - $L=\{bn \mid n=i^2 \text{ and } i>1\}$
  - $L=\{WWR \mid W \text{ is } (a,b)^*\}$
  - $L=\{anbn+1 \mid n \geq 1\}$

#### SKILLS:

- ✓ Investigate syntax and semantics of a regular and context free languages.
- ✓ Develop the problem understanding solving ability.
- ✓ Design optimized solutions for a language.

- Convert the following language or PDA to CFG
  - (a)  $L = \{anbn \mid n \geq 1\}$  and
  - (b)  $\delta(q, 0, z) = \{(q, xz)\}$
  - $\delta(q, 0, x) = \{(q, xx)\}$
  - $\delta(q, 1, x) = \{(q, x)\}$
  - $\delta(q, \cdot, x) = \{(p, \cdot)\}$
  - $\delta(p, 1, x) = \{(p, xx)\}$
  - $\delta(p, \cdot, x) = \{(p, \cdot)\}$
  - $\delta(p, 1, z) = \{(p, \cdot)\}$
- Construct PDA for the following languages
  - a)  $L = \{0^n 1^m \mid n \geq m\}$
  - b)  $L = \{anbn \mid n \geq 1\}$
  - c)  $L = \{w \mid w \in \{a, b\}^*\}$
  - d)  $L = \{w \mid na(w) > nb(w)\}$
  - e)  $L = \{0^n 1^{2n} \mid n > 0\}$
  - f)  $L =$  Where  $wR$  is reverse of  $w$
  - g)  $L =$  Where  $wR$  is reverse of  $w$
- Construct PDA for the following languages
  - a)  $L = \{anbncn \mid n > 1\}$
  - b)  $L = \{anbmambn \mid n, m \geq 1\}$
- Find the CFG corresponding to PDA whose transition mapping is as follows:
- Construct Turing Machine for the following languages
  - a)  $L = \{anbn \mid n \geq 1\}$
  - b)  $L = \{0^2 n 1^n \mid n > 0\}$
  - c)  $L = \{ww^r \mid w \text{ is } (0+1)^*\}$
  - d)  $L = \{anbncn \mid n \geq 1\}$ .
  - e) Well balanced Parenthesis for example:  $()()$

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                                                                                  | Blooms Level           | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------|------------------|
| 1      | Design abstract models of computing, including Deterministic Finite Automata (DFA), non-deterministic Finite Automata (NFA), Push Down Automata (PDA) and Turing Machine (TM) models and their power to recognize the languages. | Apply                  | 1          | 1, 3             |
| 2      | Design different finite state machines to perform various operations.                                                                                                                                                            | Apply, Analyze, Design | 1, 2       | 1, 2, 3          |
| 3      | Analyze, the given language is regular or not regular, CFL or not, Ambiguous unambiguous, Recursive and Recursive Enumerable.                                                                                                    | Analyze                | 1, 2       | 2                |
| 4      | Design Regular grammar and context free grammars for a language.                                                                                                                                                                 | Apply, Design          | 1, 2       | 1, 3             |

**TEXT BOOK:**

1. Hopcroft and Ullman, "Introduction to Automata Theory, Languages and Computation", 2nd edition, Pearson/Prentice Hall India, 2007.

**REFERENCE BOOKS:**

1. Zed A Shaw, Learn C the Hard Way: Practical Exercises on the Computational Subjects You Keep Avoiding (Like C), Addison Wesley, 2015.
2. Christoph Dürr, Sorbonne University, Jill-Jênn Vie, Inria, Competitive programming in Python, Cambridge University Press, 2020.
3. Michael Sipser, "Introduction to Theory of Computation", 3rd Edition, Course Technology, 2012.

# 22TP302 QUANTITATIVE APTITUDE AND LOGICAL REASONING

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 1 | 2 | 0 | 2 |

QUANTITATIVE  
APTITUDE

AND

LOGICAL  
REASONING

Source: <https://images.app.goo.gl/kvtVgA8TkVDCqLhj7>

**PREREQUISITE KNOWLEDGE:** Basic logical thinking and Problem solving ability.

## COURSE DESCRIPTION AND OBJECTIVES:

The Students will be introduced to various Arithmetic and Reasoning Problems. The students will have acquaintance with various problems like Time & Work, Time & distance, Percentages, Profit & Loss etc. besides solving puzzles and Critical Reasoning.

### MODULE-1

#### UNIT-1

**4L+8T+0P=12 Hours**

Number system, LCM & HCF of numbers, Percentage, Ratio and proportion, Profit, loss and discount, Average & Mixtures, Simple Interest & Compound interest.

#### UNIT-2

**4L+8T+0P=12 Hours**

Time and work, Time & distance, Problems on trains, Problems on ages, Permutation & Combinations, Probability.

#### PRACTICES:

- Each concept would be taught in detail in the class followed by 10 problems solved in the class.
- Students would have to solve 10 additional problems as a homework assignment in each concept.

### MODULE-2

#### UNIT-1

**4L+8T+0P=12 Hours**

Number series, Letter series, Analogy, Odd man out, Coding and decoding, Syllogisms- Statement & Conclusions, Puzzle test.

#### UNIT-2

**4L+8T+0P=12 Hours**

Blood relations, Direction sense test, Order & Ranking, Seating Arrangements, Calendar & Clocks.

#### PRACTICES:

- Each concept would be taught in detail in the class followed by 10 problems solved in the class. Students would have to solve 10 additional problems as homework assignments in each concept.

**SKILLS:**

- ✓ *Helps in developing and improving problem-solving skills.*
- ✓ *Flexing and honing logical abilities.*
- ✓ *Allow students to develop critical thinking skills.*

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                  | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Meet the demands of current job market besides equipping them higher studies like CAT, GMAT etc. | Apply        | 1          | 2, 5             |
| 2      | Solve Arithmetic and Reasoning Problems within shortest possible time without paper work.        | Apply        | 1          | 2, 5             |
| 3      | Exhibit better analytical skills and aptitude skills.                                            | Analyse      | 2          | 2, 4             |
| 4      | Develop interpretational skills.                                                                 | Evaluation   | 2          | 2, 4             |

**TEXT BOOKS:**

1. R. S. Aggarwal, "Quantitative Aptitude for Competitive Examinations", S. CHAND Publications- Revised Edition, 2017.
2. ARIHANT, "A New Approach to Verbal & Non-Verbal Reasoning", Arihant Publication- Revised Edition, 2021.

**REFERENCE BOOKS:**

1. Trishna Knowledge Systems, "Quantitative Aptitude for Competitive Examinations", Pearson Publication, 2013.
2. R. S. Aggarwal, "A Modern Approach to Verbal & Non-Verbal Reasoning", Revised Edition, S. CHAND Publications, 2018.



# 22AM304 FUNDAMENTALS OF IMAGE PROCESSING

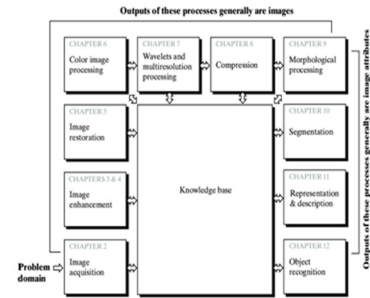
Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 2 | 3 |

**PREREQUISITE KNOWLEDGE:** Probability & Statistics.

## COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on imparting knowledge about the aspects of Image Processing and its applications. The main objective of the course is to learn digital image fundamentals, image transforms, image enhancement, restoration and compression, morphological image processing, representation and description.



Source: <https://snabnetworking.com/what-is-computer-network-and-its-types/>

## MODULE-1

### UNIT-1

**8L+0T+8P=16 Hours**

#### FUNDAMENTALS OF IMAGE PROCESSING:

Fundamental steps in digital image processing, Components of image processing system, A simple image formation model, Image sampling and quantization, Basic relationships between pixels, Introduction to Fourier Transform and DFT—properties of 2D Fourier Transform, FFT.

### UNIT-2

**8L+0T+8P=16 Hours**

#### IMAGE ENHANCEMENT:

**Basic Gray:** level transformations, Histogram processing, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters.

**Image Segmentation:** Fundamentals, Point, Line and edge detection, thresholding, Region-based segmentation, Segmentation using morphological water sheds, the use of motion in segmentation.

#### PRACTICES:

- Develop a module to enhance the image by using image arithmetic and logical operations.
- Develop a module for an image enhancement using kernel operations.
- Develop a module for gray level slicing with and without background.
- Develop a module for image enhancement using histogram equalization.
- Develop a module to filter an image using low pass & high pass filter in spatial domain. Compare the performance of both filters.
- Develop a module for smooth an image using low pass & high pass filters in frequency domain. Compare the performance of both filters.
- Develop a module for detecting lines & edges in an image.
- Develop a module for segmenting region of interest.

## MODULE-2

### UNIT-1

**8L+0T+8P = 16 Hours**

#### IMAGE RESTORATION:

A model of image degradation/restoration, Noise models, inverse filtering, wiener filtering, Constrained Least Squares Filtering, Geometric Mean Filter.

**Image Compression:** Fundamentals, Huff man coding, Golomb coding, LZW coding, Run-length coding.

**SKILLS:**

- ✓ Apply knowledge of science and engineering principles to image related problems.
- ✓ Undertake image problem identification and formulate solutions.
- ✓ Implement algorithms for enhancement, restoration, compression etc.

**UNIT-2****8L+0T+8P=16 Hours****MORPHOLOGICAL IMAGE PROCESSING:**

Erosion, Dilation, Opening, Closing, the hit-or-miss transformation; Basic morphological algorithms-boundary extraction, hole filling, extraction of connected components, thinning, thickening, skeletons, pruning.

**Feature Extraction:** Background, Boundary preprocessing, Boundary Feature Descriptors, Region Feature Descriptors, Principle Components as feature descriptors, Whole-image features.

**PRACTICES:**

- Develop a module to perform add & removal of salt and pepper noise. Compute PSNR & MSE and check the impact before and after removal of noise.
- Develop a module to remove noise using average filter and median filter. Compute PSNR & MSE before and after removal of noise.
- Develop a module for image compression and decompression.
- Develop a module for morphological image operations -erosion, dilation, opening & closing.
- Develop a module for morphological image operations - hit-or-miss transformation
- Develop a module formorphologicalimageoperations - thinning, thickening
- Develop a module for extracting boundary features of an image.
- Develop a module for extracting features of an image using GLCM.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                             | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Understand the fundamental concepts of a digital image processing system                                    | Understand   | 1          | 1, 2             |
| 2      | Learn different techniques employed for the enhancement of images.                                          | Analyse      | 1          | 1, 2, 3, 5, 12   |
| 3      | Employ image segmentation and representation techniques to extract region of interest                       | Apply        | 1          | 1, 2, 3, 5, 12   |
| 4      | Learn different causes for image degradation and overview of image restoration techniques.                  | Evaluate     | 2          | 1, 2, 3, 5, 12   |
| 5      | Apply various compression techniques to reduce image size and morphological operations to extract features. | Apply        | 2          | 1, 2, 3, 5, 12   |
| 6      | Learn different feature extraction techniques for image analysis and recognition                            | Apply        | 2          | 1, 2, 3, 5, 12   |

**TEXT BOOK:**

1. Rafeal C.Gonzalez and Richard E.Woods, "Digital Image Processing", 4thEdition, Pearson Education/PHI, 2018.

**REFERENCE BOOKS:**

1. Milanson ka, Vaclav Hlavac and Roger Boyle, "Image Processing, Analysis and Machine Vision", 4thEdition, C engage, 2015.
2. A lasdair Mc Andrew, "Introduction to Digital Image Processing with Mat lab", Thomson Course Technology, 2004 Course Technology Press, Boston, MA, United States, 2004.
3. William K. Prat, "Digital Image Processing", 4thEdition, Wiley-Inter science, A John Wiley & Sons, Inc., Publication, 2007.

**22AM305 REINFORCEMENT LEARNING**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 0 | 3 |

**PREREQUISITE KNOWLEDGE:** Solid understanding of Machine Learning theoretical and practical knowledge of Deep Learning, Proficiency in Linear Algebra, Probability and Statistics, Proficiency in Python Programming.

**COURSE DESCRIPTION AND OBJECTIVES:**

The objective of the course is to provide exposure to these advances and facilitate in depth discussions on chosen topics. The course requires that students have done the introduction to reinforcement learning course.

**MODULE-1****UNIT-1****8L+8T+0P=16 Hours****INTRODUCTION:**

Introduction and Basics of RL, Multi-armed bandits, Defining RL Framework and Markov Decision Process Policies, Value Functions and Bellman Equations, Exploration vs. Exploitation, Code Standards and Libraries used in RL (Python/Keras/TensorFlow).

**UNIT-2****8L+8T+0P=16 Hours****TABULAR METHODS AND Q-NETWORKS:**

Tabular based solutions (dynamic programming, Monte Carlo, temporal-difference), Linear value function approximation, On-linear value function approximation (Deep Q-networks (DQN, DDQN, Duelling DQN, Prioritised Experience Replay).

**PRACTICES:**

- Illustration of Various Algorithms in reinforcement learning:
  - a) Q-Learning, Policy Iteration, Value Iteration
  - b) State-Action-Reward-State-Action (SARSA)
  - c) Deep Q Network (DQN)

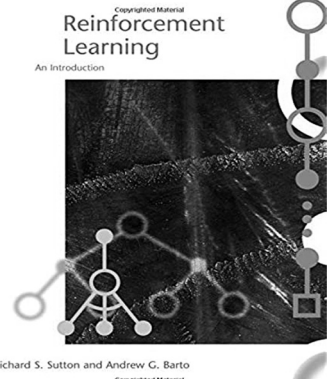
**MODULE-2****UNIT-1****8L+8T+0P=16 Hours****POLICY OPTIMIZATION:**

Introduction to policy-based methods, Vanilla Policy Gradient, REINFORCE algorithm and stochastic policy search, Actor-critic methods (A2C, A3C), Advanced policy gradient (PPO, TRPO, DDPG).

**UNIT-2****8L+8T+0P=16Hours****MODEL BASED RL:**

Bayesian Approaches to Reinforcement Learning; Data-efficient Reinforcement Learning; Learning with off-line data; Learning with incompletely specified models; RL and planning.

**Recent Advances and Applications:** Meta-learning, Multi-Agent Reinforcement Learning, Partially Observable Markov Decision Reassimilation learning (behavioural cloning), Ethics & safety in AI.



Source: <https://www.amazon.in/Reinforcement-Learning-Introduction-Richard>

**SKILLS:**

- ✓ Analyze the intelligent system.
- ✓ Apply real-time problem-solving techniques.
- ✓ Explore imitation learning tasks and solutions.

**PRACTICES:**

- Asynchronous Advantage Actor-Critic Algorithm(A3C).
- Proximal Policy Optimization Algorithm.
- Soft Actor-Critics(SAC) Algorithm.

**CASE STUDY:**

- Playing games like Go.
- Self-driving cars.
- Customized action in video games.
- Samuel's Checker player.
- Customized action in video games.
- Job-Stop scheduling.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                               | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply RL search models and search strategies for problem solving.             | Apply        | 1          | 1, 2, 12         |
| 2      | Analyze the logic for representing knowledge and reasoning for RL systems.    | Analyze      | 1          | 1, 2, 12         |
| 3      | Design the different Reinforcement algorithms for real time problems.         | Design       | 1          | 1, 2, 3, 5, 12   |
| 4      | Evaluate the searching strategies to achieve the goals for certain situation. | Evaluate     | 2          | 1, 2, 12         |
| 5      | Conduct investigation and implement projects using RL techniques.             | Create       | 2          | 1, 2, 5, 12      |

**TEXT BOOKS:**

1. Richard S. Sutton and Andrew G. Barto, "Reinforcement learning: An introduction", Second Edition, MIT Press, 2019.
2. Neuro-Dynamic Programming. Dimitri Bertsekas and John G. Tsitsiklis. "Athena Scientific", 1996.

**REFERENCE BOOKS:**

1. Wiering, Marco, and Martijn Van Otterlo. "Reinforcement learning. Adaptation, learning, and optimization", 2012.
2. Russell, Stuart J., and Peter Norvig. "Artificial intelligence: a modern approach", Pearson Education Limited, 2016.
3. Goodfellow, Ian, YoshuaBengio, and Aaron Courville. "Deep learning." MIT press, 2016.

**22CS303 WEB TECHNOLOGIES**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 4 | 4 |



Source: <https://www.dreamstime.com/web-development-coding-programming-internet-technology-business-concept-web-development-coding-programming-internet-technology-image121903546>

**PREREQUISITE KNOWLEDGE:** OOPs through JAVA.

**COURSE DESCRIPTION AND OBJECTIVES:**

This course offers the concepts of web development like static and dynamic web page design and provides internet programming knowledge, web servers, application servers, and design methodologies using object-oriented concepts. The objective of this course is to build web applications using JSP, PHP, Angular JS, and Node JS with a client and server-side scripting technologies that span multiple domains.

**MODULE-1****UNIT-1****8L+0T+16P=24 Hours****INTRODUCTION:**

**HTML:** Creating structured documents, Links and navigation, Tables, Forms, and Frames.

**HTML 5:** Introduction to HTML5, The HTML5 Canvas, HTML5 audio and Video;

**CSS:** Cascading Style Sheets, CSS Properties.

**Java Script:** Learning Java Script- how to add scripts to your page, DOM, variables, operators, functions, conditional statements, Looping, Events, Built-in objects, form and regular expression validation.

**UNIT-2****8L+0T+16P=24 Hours****JDBC AND JSP:**

**JDBC:** What is JDBC, system requirements, types of JDBC Drivers, creating database tables, connecting to a database, executing SQL statements, processing result sets, and making changes to a result set.

**JSP:** JSP Processing, Generating Dynamic Content using Scripting Elements, Implicit JSP Objects, Sharing Data between JSP pages, JSP application design with JDBC.

**PRACTICES:**

- Design a webpage having four frames named a)Top, b)Center, c)Bottom, and d) Left. The top frame should contain the company logo and title. The bottom frame should contain copyright information. The left frame should contain various links like Home, Products, Services, Branches, About, etc., When clicked on respective links, the content should display on the center frame.
- Design a catalog page that should contain the details of all the books available on the website in a table. The details should contain the following: a) Snapshot of Cover Page b) Author Name c) Publisher. d) Price. e) Add to cart button.
- Design a timetable schedule for your current semester using the Table tag.
- Design a HTML page for Student Registration Form using Form Elements that includes Form, input-text, password, radio, checkbox, hidden, button, submit, reset, label, textarea, select, option and file upload.
- Design a HTML web page with at least two <h1>, two images, two buttons, and appropriate CSS to display,  
All<h1>withfont-size12pt, and bold in Verdana font using In line CSS.  
All<img>withbordercoloryellow,thickness10pxusingDocumentLevelCSS.  
All<inputtype='button'>shouldchangebackgroundcolortoredonmouseoverthemusing External CSS.

- Design a HTML page having a text box and four buttons viz Factorial, Fibonacci, Prime and Palindrome. When a button is pressed an appropriate java script function should be called to display the following:  
Factorial of that number.  
Fibonacci series up to that number.  
Prime numbers up to that number.  
Is it palindrome or not?
- Design a web page that contains a color pallet, when the user moves the mouse to the particular area, then it changes the background color of the web page.
- Design a registration page to validate the following fields using Java Script.  
Make sure the user name starts with an upper case letter.  
The user name must have at least one digit.  
Ensure that Email is valid.  
Ensure that the password length is between 8 to 20 characters.  
Make sure the password contains at least one upper case letter, one lower case, and one special character exclude [ . (dot), ,(comma), ;(semicolon), : (colon)].
- Design a web page to display the videos on-page, on user selection using frames and HTML5 tags.
- Design a web page to display different types of objects using HTML5 Canvas.
- Design a web application to validate entered username and password through JDBC connection program and display user information on successful login and provide profile editing option to the user. Else display an error message.
- Develop a JSP application to create a user on successful signup and update user information on successful login and display user information on the home screen and provide a logout button.
- Make an HTML form that collects the last name. Send the name to JSP page. If there is an employee with that last name, show full details of him or her (just show the first employee if there are multiple people with the same name). If there is no employee with that last name, say “no employee records available.”

## MODULE-2

### UNIT-1

8L+0T+16P=24 Hours

#### PHP:

Introduction to PHP, Expressions, and control flow in PHP, functions and objects, Arrays, Accessing MySQL using PHP, Form Handling, Cookies, Sessions, and Authentication.

### UNIT-2

8L+0T+16P=24 Hours

#### ANGULAR JS AND NODE JS:

**Angular JS:** Introduction, Expressions, Modules, Directives, Controllers, Filters, Events, Forms, Form Validation.

**Node JS:** Introduction, Setup Dev Environment, Modules, Node Package Manager, Creating Web server, File System, Events, Express.js, Accessing MySQL from Node.js.

#### PRACTICES:

- Design a web page using PHP, upload image into web page and display image, when user clicking on view button.
- Design a personal Information form, Submit & Retrieve the form data using `$_GET()`, `$_POST()` and `$_REQUEST()` Variables.
- Design a login page to validate username and password through MySQL. If login is successful display user information on home page and modify user information on edit page using sessions. When user logged out, destroy all user-related sessions.

- Design a web page to accept payment data from user and do the payment, on successful payment display details on the screen. A Session should be set while doing payment up to 10 minutes after that link/payment page should be destroyed irrespective of user payment.
- Design a web application to validate user registration page using Angular JS.
- Design a search engine using Angular JS. On key press, display data on web page.
- Design a web page to validate user name and password using: Node.js and PHP. When user clicks on login button, server checks the data availability in data base. If the data matches a successful login page is returned. Otherwise, a failure message is shown to the user.
- Design a web application to display the active duration of the user, i.e., time between login and logout.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                             | Blooms Level | Module No. | Mapping with POs |
|--------|-----------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Usage of HTML, HTML5, CSS, Java Script, and PHP in web application development.                                             | Apply        | 1, 2       | 1                |
| 2      | Apply Angular JS features for form validation and Node JS, and JDBC concepts to perform database operations from web pages. | Apply        | 1, 2       | 1                |
| 3      | Analyse the suitability of Node JS and JSP technologies to build solutions for real-world problems.                         | Analyse      | 2          | 2                |
| 4      | Design and develop three tier web applications using JSP, Node JS, Angular JS, and PHP.                                     | Creating     | 2          | 3                |

**TEXT BOOKS:**

1. Jon Duckett, "Beginning Web Programming with HTML, XHTML, and CSS", 2nd Edition, Wiley Publishing, Inc, 2008.
2. Robin Nixon, "Learning PHP, MySQL & JavaScript WITH JQUERY, CSS & HTML5", 4th Edition, O'Reilly, 2015.

**REFERENCEBOOKS:**

1. Paul Deitel, Harvey Deitel, Abbey Deitel, "Internet & World Wide Web How to Program", 5th Edition, Pearson Education, 2012.
2. Kishori Sharon, "Java APIs, Extensions and Libraries with JavaFX, JDBC, jmod, jlink, Networking and the process API", 2nd Edition, Apress, 2018.
3. Brad Dayley, Brendan Dayley, and Caleb Dayley, "Node.js, Mongo DB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications," 2nd Edition, Pearson Education, 2018.
4. Steve Prettyman, "Learn PHP 7 Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL", 1st edition, Apress, 2015.
5. Adrian W. West and Steve Prettyman, "Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites", 1st edition, A Press, 2018.





**IV**  
YEAR

**B.Tech.**

# CSE - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

## I SEMESTER

- ▶ 22AM401 - Knowledge Representation and Reasoning
- ▶ 22AM402 - Text Mining
- ▶ 22CS402 - Big Data Analytics
- ▶ - Department Elective - 3
- ▶ - Department Elective - 4

## II SEMESTER

- ▶ 22AM403 - Internship / Project Work

**COURSE CONTENTS**

I SEM & II SEM



# 22AM401 KNOWLEDGE REPRESENTATION AND REASONING

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 0 | 3 |



Source: [https://medium.com/@mdsaife245/knowledge-representation-and-reasoning-chapter-1-basic-gyaan-about-knowledge-representation-7630fdaa5a53?source=rss-----artificial\\_intelligence-5](https://medium.com/@mdsaife245/knowledge-representation-and-reasoning-chapter-1-basic-gyaan-about-knowledge-representation-7630fdaa5a53?source=rss-----artificial_intelligence-5)

**PREREQUISITE KNOWLEDGE:** Artificial Intelligence.

## COURSE DESCRIPTION AND OBJECTIVES:

An intelligent agent needs to be able to solve problems in its world. The ability to create representations of the domain of interest and reason with these representations is a key to intelligence. In this course we explore a variety of representation formalisms and the associated algorithms for reasoning. We start with a simple language of propositions, and move on to first order logic, and then to representations for reasoning about action, change, situations, and about other agents in incomplete information situations. This course is a companion to the course Artificial Intelligence: Search Methods for Problem Solving that was offered recently and the lectures for which are available online.

## MODULE-1

### UNIT-1

**6L+6T+0P=12 Hours**

#### INTRODUCTION TO KNOWLEDGE REPRESENTATION & REASONING:

An Introduction to Formal Logics, Propositional Logic: Language, Semantics and Reasoning, Syntax and Truth Values, Valid Arguments and Proof Systems, Rules of Inference and Natural Deduction, The Resolution Refutation Method.

### UNIT-2

**10L+10T+0P=20 Hours**

#### THE LANGUAGE OF FIRST ORDER LOGIC:

First Order Logic (FOL), Syntax and Semantics, Entailment and Models, Proof Systems, Forward Chaining, Unification, Forward Chaining Rule Based Systems. The Rete Algorithm, Rete example, Programming Rule Based Systems, Representation in FOL.

#### PRACTICES:

- Consider the following piece of knowledge:  
Tony, Mike, and John belong to the Alpine Club. Every member of the Alpine Club who is not a skier is a mountain climber. Mountain climbers do not like rain, and anyone who does not like snow is not a skier. Mike dislikes whatever Tony likes, and likes whatever Tony dislikes. Tony likes rain and snow.
  - Prove that the given sentences logically entail that there is a member of the Alpine Club who is a mountain climber but not a skier.
  - Suppose we had been told that Mike likes whatever Tony dislikes, but we had not been told that Mike dislikes whatever Tony likes. Prove that the resulting set of sentences no longer logically entails that there is a member of the Alpine Club who is a mountain climber but not a skier.
- Consider the following facts about the Elm Street Bridge Club:  
Joe, Sally, Bill, and Ellen are the only members of the club. Joe is married to Sally. Bill is Ellen's brother. The spouse of every married person in the club is also in the club. From these facts, most people would be able to determine that Ellen is not married.
  - Represent these facts as sentences in FOL, and show semantically that by themselves they do not entail that Ellen is not married.
  - Write in FOL some additional facts that most people would be expected to know, and show that the augmented set of sentences now entails that Ellen is not married.

**SKILLS:**

- ✓ Converse with people in natural languages.
- ✓ Determine the course of action.
- ✓ Plan future activities.
- ✓ Perform human-like problem-solving.

- Donald and Daisy Duck took their nephews, age 4, 5, and 6, on an outing. Each boy wore a tee-shirt with a different design on it and of a different color. You are also given the following information:
  - I. Huey is younger than the boy in the green tee-shirt.
  - II. The 5-year-old wore the tee-shirt with the camel design.
  - III. Dewey's tee-shirt was yellow.
  - IV. Louie's tee-shirt bore the giraffe design.
  - V. The panda design was not featured on the white tee-shirt.
  - a) Represent these facts as sentences in FOL.
  - b) Using your formalization, is it possible to conclude the age of each boy together with the color and design of the tee-shirt he is wearing? Show semantically how you determined your answer.
  - c) If your answer was "no," indicate what further sentences you would need to add so that you could conclude the age of each boy together with the color and design of the tee-shirt he is wearing. √
- Prove each of the following assertions:
  - a)  $\alpha$  is valid if and only if  $\text{True} \models \alpha$ .
  - b) For any  $\alpha$ ,  $\text{False} \models \alpha$ .
  - c)  $\alpha \models \beta$  if and only if the sentence  $(\alpha \Rightarrow \beta)$  is valid.
  - d)  $\alpha \equiv \beta$  if and only if the sentence  $(\alpha \Leftrightarrow \beta)$  is valid.
  - e)  $\alpha \models \beta$  if and only if the sentence  $(\alpha \wedge \neg\beta)$  is unsatisfiable.
- Prove, or find a counterexample to, each of the following assertions:
  - a) If  $\alpha \models \gamma$  or  $\beta \models \gamma$  (or both) then  $(\alpha \wedge \beta) \models \gamma$
  - b) If  $\alpha \models (\beta \wedge \gamma)$  then  $\alpha \models \beta$  and  $\alpha \models \gamma$ .
  - c) If  $\alpha \models (\beta \vee \gamma)$  then  $\alpha \models \beta$  or  $\alpha \models \gamma$  (or both).
- Describe the wumpus world according to the properties of task environment
- Represent the following sentences in first-order logic, using a consistent vocabulary(which you must define):
  - a) Some students took French in spring 2001.
  - b) Every student who takes French passes it.
  - c) Only one student took Greek in spring 2001.
  - d) The best score in Greek is always higher than the best score in French.
  - e) Every person who buys a policy is smart.
  - f) No person buys an expensive policy.

**MODULE-2****UNIT-1****10L+10T+0P=20 Hours****RESOLUTION AND REASONING WITH DESCRIPTION LOGIC**

Deductive Retrieval, Backward Chaining, Logic Programming, Resolution Refutation in FOL, FOL with Equality, Complexity of Theorem Proving, Description Logic (DL), Structure Matching, Classification, Extensions of DL, The ALC Language, Inheritance in Taxonomies, Default Reasoning, Circumscription.

**UNIT-2****6L+6T+0P=12 Hours****DEFAULTS**

Default Logic, Auto epistemic Logic, Epistemic Logic, The Muddy Children Puzzle, Multi Agent Scenarios, Conceptual Dependency (CD) Theory, Understanding Natural Language.

**PRACTICES:**

- Consider the following axioms:

- a) Every child loves Santa.
- b) Everyone who loves Santa loves any reindeer.
- c) Rudolph is a reindeer, and Rudolph has a red nose.
- d) Anything which has a red nose is weird or is a clown.
- e) No reindeer is a clown.
- f) Scrooge does not love anything which is weird.

Apply resolution technique to conclude Scrooge is not a child.

- Consider the following axioms:
  - a) John likes all kind of food.
  - b) Apple and vegetable are food
  - c) Anything anyone eats and not killed is food.
  - d) Anil eats peanuts and still alive
  - e) Harry eats everything that Anil eats.

Prove by resolution that: John likes peanuts.

- Consider the following axioms:
  - a) Anyone who buys carrots by the bushel owns either a rabbit or a grocery store.
  - b) Every dog chases some rabbit.
  - c) Mary buys carrots by the bushel.
  - d) Anyone who owns a rabbit hates anything that chases any rabbit.
  - e) John owns a dog.
  - f) Someone who hates something owned by another person will not date that person.

Prove by resolution that If Mary does not own a grocery store, she will not date John.

- Consider the following axioms:
  - a) Anyone whom Mary loves is a football star.
  - b) Any student who does not pass does not play.
  - c) John is a student.
  - d) Any student who does not study does not pass.
  - e) Anyone who does not play is not a football star.

Prove by resolution that If John does not study, then Mary does not love John.

- Consider the following axioms:
  - a) Every coyote chases some roadrunner.
  - b) Every roadrunner who says ``beep-beep'' is smart.
  - c) No coyote catches any smart roadrunner.
  - d) Any coyote who chases some roadrunner but does not catch it is frustrated.

Prove by resolution that If all roadrunners say ``beep-beep'', then all coyotes are frustrated.

- Consider the following axioms:
  - a) Anyone who rides any Harley is a rough character.
  - b) Every biker rides [something that is] either a Harley or a BMW.
  - c) Anyone who rides any BMW is a yuppie.
  - d) Every yuppie is a lawyer.
  - e) Any nice girl does not date anyone who is a rough character.
  - f) Mary is a nice girl, and John is a biker.

Prove by resolution that If John is not a lawyer, then Mary does not date John.

- Consider the following axioms:
  - a) Every child loves anyone who gives the child any present.
  - b) Every child will be given some present by Santa if Santa can travel on Christmas eve.
  - c) It is foggy on Christmas eve.
  - d) Anytime it is foggy, anyone can travel if he has some source of light.
  - e) Any reindeer with a red nose is a source of light.

Prove by resolution that If Santa has some reindeer with a red nose, then every child loves Santa.

- Consider the following axioms:
  - a) Every investor bought [something that is] stocks or bonds.
  - b) If the Dow-Jones Average crashes, then all stocks that are not gold stocks fall.
  - c) If the T-Bill interest rate rises, then all bonds fall.
  - d) Every investor who bought something that falls is not happy.

Prove by resolution that If the Dow-Jones Average crashes and the T-Bill interest rate rises, then any investor who is happy bought some gold stock.

- Consider the following axioms:
  - a) Every child loves every candy.
  - b) Anyone who loves some candy is not a nutrition fanatic.
  - c) Anyone who eats any pumpkin is a nutrition fanatic.
  - d) Anyone who buys any pumpkin either carves it or eats it.
  - e) John buys a pumpkin.
  - f) Lifesavers is a candy.

Prove by resolution that If John is a child, then John carves some pumpkin.

- Consider the following axioms:
  - a) Every tree that is an oak contains some grackle.
  - b) If anyone walks under any tree that contains any grackle, then he hates every grackle.
  - c) For every building, there is some tree that is beside it.
  - d) Taylor Hall is a building.
  - e) Every CS student visits Taylor Hall.
  - f) If anyone visits any building, then he walks under every tree that is beside that building.

Prove by resolution that If some CS student does not hate some grackle, then there is some tree beside Taylor Hall that is not an oak.

- Consider the following axioms:
  - a) Every child sees some witch.
  - b) No witch has both a black cat and a pointed hat.
  - c) Every witch is good or bad.
  - d) Every child who sees any good witch gets candy.
  - e) Every witch that is bad has a black cat.

Prove by resolution that If every witch that is seen by any child has a pointed hat, then every child gets candy.

- Consider the following axioms:
  - a) Every boy or girl is a child.
  - b) Every child gets a doll or a train or a lump of coal.
  - c) No boy gets any doll.
  - d) No child who is good gets any lump of coal.
- Prove by resolution that If no child gets a train, then no boy is good.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | To create an intelligent machine, which makes it a crucial part of artificial intelligence.                                                                    | Apply        | 1          | 1,3              |
| 2      | Inspect and analyze Logic for representing Knowledge and Reasoning of AI systems and Conduct investigation and implement project using AI learning techniques. | Analyze      | 1          | 2                |
| 3      | Apply techniques specific to specific knowledge representation problems and formalisms.                                                                        | Apply        | 2          | 6                |
| 4      | Develop of theoretical and practical issues in symbolic knowledge representation and reasoning, in general.                                                    | Create       | 2          | 3                |

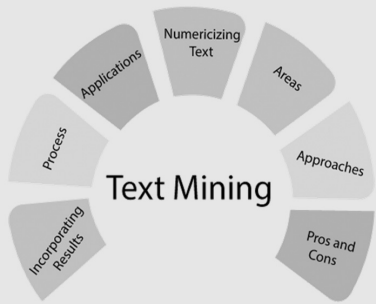
**TEXT BOOKS:**

1. Ronald J. Brachman, Hector J. Levesque, "Knowledge Representation and Reasoning", Morgan Kaufmann, 2018.
2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education (India), 2013.

**REFERENCEBOOKS:**

1. Schank, Roger C., Robert P. Abelson, "Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures", Psychology Publishes, 2013.
2. R. C. Schank and C. K. Riesbeck, "Inside Computer Understanding: Five Programs Plus Miniatures", Taylor and Francies, 1981.
3. John F. Sowa, "Knowledge Representation: Logical, Philosophical, and Computational Foundations", Thomson Learning, 2000.

# 22AM402 TEXT MINING



Source: <https://www.javatpoint.com/text-data-mining>

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Data analysis skills, Database, Machine Learning or Deep Learning Algorithms.

## COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the fundamental concepts and techniques used in Text Processing. Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information. The course examines models and algorithms used in both the traditional symbolic and the more recent statistical approaches.

## MODULE-1

### UNIT-1

**10L+0T+6P=16 Hours**

#### INTRODUCTION

NLP tasks in syntax, semantics, and pragmatics. Applications such as information extraction, question answering, and machine translation. The role of machine learning. Brief history of the field.

**Regular Expressions, Text Normalization, Edit Distance:** Regular Expressions, words, Corpora, Text Normalization and Minimum edit distance.

### UNIT-2

**14L+0T+10P=24 Hours**

#### N-GRAMS, VECTOR SEMANTICS AND EMBEDDING

**N-Gram Language Models:** N-Grams, Evaluating Language Models, Generalization and Zeros, Smoothing, Kneser-Ney Smoothing, Huge Language Models and Stupid Back off.

**Vector Semantics and Embeddings:** Lexical semantics, Vector semantics, words and vectors, cosine for measuring similarity, TF-IDF: weighing terms in the vector, Point wise Mutual Information, Applications of TF/ IDP and PPMI vector models, word 2 vec, visualizing embeddings, semantic properties of embeddings, bias and embeddings, evaluating vector models.

#### PRACTICES:

- Perform basic text pre-processing using the following approaches:
  - Stop word Elimination.
  - Removal of Special Characters.
  - Stemming.
  - Lemmatization.
  - N-gram Tokenization.
- Design and develop a text classification model with various machine learning algorithms over the following feature extraction methods and compare their performance.
  - TF-IDF.
  - Word2Vec.
- Perform Exploratory Data Analysis using following:
  - Word Cloud.
  - Frequent Word Detection.
  - Keyword Extraction based on ranking.



**MODULE-2****UNIT-1****12L+0T+8P=20 Hours****SEQUENCE LABELING FOR PARTS OF SPEECH AND NAMED ENTITIES**

Part-of-Speech Tagging, Named Entities and Named Entity Tagging, HMM Part-of-Speech Tagging, Conditional Random Fields, Evaluation of Named Entity Recognition.

**UNIT-2****12L+0T+8P=20 Hours****SEQUENCE AND TRANSLATION PROCESSING**

**Deep Learning Architectures for Sequence Processing:** Language Models Revisited, Self-Attention Networks: Transformers.

**Machine Translation and Encoder-Decoder Models:** The Encoder-Decoder Model, Encoder-Decoder with RNNs, Attention, Beam Search, Encoder-Decoder with Transformers.

**PRACTICES:**

- Perform Named Entity Recognition to extract required entities from a given unstructured text using NLTK.
- Perform POS Tagging with HMM and also optimize the performance of HMM with Viterbi.
- Design and develop a text classification model using Latent Dirichlet allocation and compare its performance with TF-IDF and Word2Vec.
- Perform text classification using following methods and compare their performance in terms of various evaluation metrics such as Accuracy, Precision, Recall and F-Score.
  - CNN
  - LSTM
  - GRU
  - Encoder-Decoder-Transformers
- Design and develop text classification model using Attention.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Understand usage of Regular Expressions to process the raw text.               | Understand   | 1          | 1,3              |
| 2      | Apply various approaches to sentiment analysis using Machine Learning methods. | Apply        | 1          | 2                |
| 3      | Analyze the vector semantics and embedding in the representation of the text.  | Analyze      | 2          | 6                |
| 4      | Design & Development various statistical approaches to machine translation.    | Implement    | 3,4,5      | 3                |

**TEXT BOOK:**

1. Daniel Jurafsky and James H Martin, "Speech and Language processing: An introduction to Natural Language Processing, Computational Linguistics and speech Recognition", 3rd edition, 2020.

**REFERENCEBOOKS:**

1. Steven Bird, Ewan Klein, Edward Loper, "Natural Language Processing with Python", 1st edition, O'Reilly Publishers, 2009.
2. Nitin Indurkha, Fred J. Damerau, "Handbook of Natural Language Processing", 2nd edition, CRC Publishers, 2010.

**SKILLS:**

- ✓ Exploratory Data analysis.
- ✓ Opinion Mining/ Sentiment Analysis using various deep Learning models.
- ✓ Text Summarization and Categorization etc.

# 22CS402 BIG DATA AND ANALYTICS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Basics of databases, Data mining.

## COURSE DESCRIPTION AND OBJECTIVES:

This course serves as an introductory course to gain knowledge on analyzing Big Data. Expecting to face Big Data storage, processing, analysis, visualization, and application issues on both workplaces and research environments. Get insight on what tools, algorithms, and platforms to use on which types of real world use cases.

## MODULE-1

### UNIT-1

12L+0T+8P=20 Hours

#### INTRODUCTION TO BIG DATA:

Data, Characteristics of data and Types of digital data, Sources of data, Working with unstructured data, Evolution and Definition of big data, Characteristics and Need of big data, Challenges of big data.

**Big data analytics:** Overview of business intelligence, Data science and Analytics, Meaning and Characteristics of big data analytics, Need of big data analytics, Classification of analytics, Challenges to big data analytics, Importance of big data analytics, Basic terminologies in big data environment.

### UNIT-2

12L+0T+8P=20 Hours

#### INTRODUCTION TO HADOOP:

Introducing Hadoop, need of Hadoop, limitations of RDBMS, RDBMS versus Hadoop, Distributed Computing Challenges, History of Hadoop, Hadoop Overview, Use Case of Hadoop, Hadoop Distributors, HDFS (Hadoop Distributed File System), Processing Data with Hadoop, Managing Resources and Applications with Hadoop YARN (Yet another Resource Negotiator), Interacting with Hadoop Ecosystem.

#### PRACTICES:

- Hadoop installation in standalone machine.
- Pig installation.
- Setup of Hadoop cluster.
- HDFS basic command-line file operations.
- HDFS monitoring User Interface.

## MODULE-2

### UNIT-1

12L+0T+8P=20 Hours

#### MAPREDUCE PROGRAMMING:

Introduction, Mapper, Reducer, Combiner, Partitioner, Searching, Sorting, Compression, Real time applications using Map Reduce, combiner, Partitioner, matrix multiplication using Map Reduce and page rank algorithm using Map Reduce.

Source: [https://miro.medium.com/max/844/0\\*ARAg3FnAzy2e02Ww.png](https://miro.medium.com/max/844/0*ARAg3FnAzy2e02Ww.png)

**UNIT-2****12L+0T+8P=20 Hours****PIG:**

**Introduction to Pig, The Anatomy of Pig, Pig on Hadoop, Pig Philosophy, Use Case for Pig:** ETL Processing, Pig Latin Overview, Data Types in Pig, Running Pig, Execution Modes of Pig, HDFS Commands, Relational Operators, Piggy Bank, Word Count Example using Pig, Pig at Yahoo!.

**Hive:** Introduction to Hive, Hive Architecture, Hive Data Types, Hive File Format, Hive Query Language (HQL), Partitions and bucketing, working with XML files, User-Defined Function (UDF) in Hive, Pig versus Hive.

**Spark Programming:** Introduction, features of Spark, components of Spark, Programming with Resilient Distributed datasets (RDDS).

**PRACTICES:**

- Word Count Map Reduce program using Hadoop.
- Implementation of word count with combiner Map Reduce program.
- Practice on Map Reduce Monitoring User Interface.
- Implementation of Sort operation using Map Reduce.
- Map Reduce program to count the occurrence of similar words in a file by using partitioner.
- Design Map Reduce solution to find the years whose average sales is greater than 30.
  - input file format has year, sales of all months and average sales.
  - Year Jan Feb Mar April May Jun July Aug Sep Oct Nov Dec Average.
- Map Reduce program to find Dept wise salary.
  - Empno Emp Name Dept Salary.
- Designing of Pig Latin scripts to sort, group, join, project and filter the data.
- Implementation of Word count using Pig.
- Creation of Database and tables using Hive query language.
- Implementation of partitions and buckets using Hive query language.
- Implementation of word count using spark RDD.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                    | Blooms Level | Module No. | Mapping with POs    |
|--------|--------------------------------------------------------------------------------------------------------------------|--------------|------------|---------------------|
| 1      | Use of Big data frameworks like Hadoop and NOSQL to efficiently store and process Big data to generate analytics.  | Apply        | 1          | 1, 2, 5,9,10,12     |
| 2      | Design a solution for data intensive problems using Map Reduce paradigm.                                           | Apply        | 1          | 1, 2, 5, 9,10,12    |
| 3      | Design and analyze the solutions of Big data using Pig and Hive to solve data intensive and to generate analytics. | Apply        | 2          | 1, 2, 3, 5, 9,10,12 |
| 4      | Analyze Big data using Spark programming.                                                                          | Analyze      | 2          | 1, 2, 3, 5, 9,10,12 |

**TEXT BOOKS:**

1. Seema Acharya, Subhashini Chellappan, "Big Data Analytics", Wiley, 2015.
2. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, "Learning Spark: Lightning-Fast Data Analysis", O'Reilly Media, Inc., 2015.

**REFERENCE BOOKS:**

1. Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich, "Professional Hadoop Solutions", Wiley, 2015.
2. Chris Eaton, Dirk deRooset al. , "Understanding Big data ", McGraw Hill, 2012.
3. Tom White, "HADOOP: The definitive Guide", O Reilly 2012.

**SKILLS:**

- ✓ *Build and maintain reliable, scalable, distributed systems with Apache Hadoop.*
- ✓ *Develop Map Reduce based applications for Big data.*
- ✓ *Design and build applications using Hive and pig based Big data applications.*
- ✓ *Learn tips and tricks for big data use cases and solutions.*



# DEPT. ELECTIVES

## B.Tech.

### COURSE CONTENTS

I SEM & II SEM

# CSE - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

#### DEPARTMENT ELECTIVES

- ▶ 22AM801 - Cloud Computing
- ▶ 22AM802 - Compiler Design
- ▶ 22AM803 - Computer Vision
- ▶ 22AM804 - Cryptography and Network Security
- ▶ 22AM805 - Data Science using Python
- ▶ 22AM806 - Data Warehousing and Data Mining
- ▶ 22AM807 - Nature Inspired Computing Methods
- ▶ 22AM808 - Soft Computing
- ▶ 22AM809 - Introduction to Software Engineering
- ▶ 22AM810 - Internet of Things
- ▶ 22CS801 - Advanced Data Structures
- ▶ 22CS802 - Advanced JAVA Programming
- ▶ 22CS803 - Computer Graphics
- ▶ 22CS808 - Mobile Application Development



# 22AM801 CLOUD COMPUTING

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Operating systems and Computer networks.

## COURSE DESCRIPTION AND OBJECTIVES:

This course deals with the computing model, which enables information, software, and shared resources to be provisioned over the network as services in an on-demand manner. The main objective of this course is to enable the student to understand the evolution of cloud computing through its supporting technologies virtualization and the architectures of top cloud platforms.

## MODULE-1

### UNIT-1

12L+0T+8P=20 Hours

#### INTRODUCTION:

**Introduction:** Definition, Historical developments, Computing platforms, and technologies.

**Principles Of Parallel and Distributed Computing:** Parallel versus distributed computing, Elements of parallel computing, Elements of distributed computing, Technologies for distributed computing.

### UNIT-2

12L+0T+8P=20 Hours

#### VIRTUALIZATION:

Introduction, Characteristics, Virtualization techniques, Virtualization and cloud computing, Pros. and cons. of virtualization, Technology examples.

**Cloud Computing Architecture:** Introduction, Cloud reference model, Types of clouds, Economics of clouds, Open challenges.

**Cloud Platforms in Industry:** Amazon web Services, Google app engine, Microsoft Azure.

#### PRACTICES:

- Performing hardware virtualization using Vmware workstation.
- Launch Amazon Linux EC2 Instance and connect the windows client to it.
- Launch Windows EC2 instance in AWS and connect windows client to it.
- Configure Web Server on Amazon Linux instance with Elastic IP.
- Manage Elastic Block Storage(EBS).
- Configure Amazon Simple Storage Service (Amazon s3).
- Configure Amazon S3 Glacier.
- Configure Amazon EFS.

## MODULE-2

### UNIT-1

12L+0T+8P=20 Hours

#### ANEKA:

Cloud application platform, Framework overview, Anatomy of the Aneka container, Building Aneka clouds, Cloud programming, and management.

High Throughput Computing- Task Programming: Task computing, Task-based application models, Aneka task-based programming.



Source: <https://tse4.mm.bing.net/th?id/OIP.vN86iZCAdr3RDSXc0cuHcAHaE8?pid=ImgDet&rs=1>

**SKILLS:**

- ✓ Gain knowledge of different types of Cloud Service Providers.
- ✓ Explore basic design issues of Cloud Applications.
- ✓ Compare & evaluate the optimum costs in the data transmissions.

**UNIT-2****12L+0T+8P=20 Hours****CLOUD APPLICATIONS:**

Scientific applications in healthcare, biology, geo science; Business applications in CRM and ERP, productivity, social networking, media applications, multiplayer online gaming.

**PRACTICES:**

- Configure Amazon Virtual Private Cloud (VPC).
  - a) Create your own VPC.
  - b) Create a public subnet.
  - c) Create a private subnet.
  - d) Create an Internet gateway and attach to your VPC.
  - e) Create Public Routing Table, associate subnet and add routing rules.
  - f) Create Private Routing Table, associate subnet and add routing Rules.
  - g) To launch Windows instance in Public subnet.
- Configure Amazon Elastic Load Balancer.
- Configure Relational Database Service (RDS).

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                      | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Deploying a VM Image to understand the evolution of cloud computing in contrast to the traditional approach.                         | Apply        | 1          | 1, 5             |
| 2      | Evaluate the concepts of various virtualization technologies.                                                                        | Evaluate     | 1          | 2, 5             |
| 3      | Analyze the trade-offs, security, and privacy issues among application deployment in the various cloud and the local infrastructure. | Analyze      | 2          | 2, 5             |
| 4      | Deploy applications over commercial cloud computing infrastructures.                                                                 | Apply        | 2          | 1, 5             |

**TEXT BOOKS:**

1. Raj Kumar Buyya, C Vecchiola and S Tselvi, "Mastering Cloud Computing", 1st Edition, Tata McGraw Hill Education (India), 2013.
2. Raj Kumar Buyya, Broberg J and GoscinskiA, "Cloud Computing - Principles and Paradigms", 1st Edition, Wiley, 2011.

**REFERENCE BOOKS:**

1. David S. Linthicum, Cloud Computing and SOA Convergence in Your Enterprise A Step-by-Step Guide, Pearson 2010.
2. Dr. Kumar Saurabh, Cloud Computing, 2nd Edition, Wiley India 2012.
3. Ritting house J W, and Ransome J F, "Cloud Computing - Implementation, Management, and Security", 1st Edition, CRC Press, 2009.
4. Michael Wittig and Andreas Wittig, "Amazon Web Services in Action", 2nd Edition, Manning Publications, 2015.
5. Tim Mather, Subra Kumaraswamy, Shahed Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly MediaInc, 2009.



# 22AM802 COMPILER DESIGN

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 4 |



Source: <https://www.javatpoint.com/compiler-tutorial>

**PREREQUISITE KNOWLEDGE:** Programming for problem solving- I & II and Formal languages and automata theory.

## COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the foundation for understanding the theory and practice of compilers and compiler design concepts; symbol table management, compiler parsing techniques, semantic analysis and optimized code generation. This course introduced the concepts of lexical analyzer, parser, code generation and code optimization techniques. The objective of this course is to enable the student to acquire the knowledge of various phases of compiler such as lexical analyzer, parser, code optimization and code generation.

## MODULE-1

### UNIT-1

**10L+6T+0P=16 Hours**

#### INTRODUCTION:

The evolution of programming languages and basic language processing system; The structure of a compiler; Bootstrapping; Lexical analyser and its Role; Input buffering; Specifications and recognition of tokens; LEX.

### UNIT-2

**14L+10T+0P=24 Hours**

#### SYNTAX ANALYSIS:

The role of the parser; Context-free grammars; Types of parsers with examples, YACC.

**Semantic Analysis:** Type checking; Syntax directed definition (SDD) and translation schemes (TS); Application of SDD and TS; Translation of expressions and control flow statements.

#### PRACTICES:

- Implement various phases of compiler in detail. Write down the output of each phase for expression  $Total = (b + c) + (b + c) * 50$ .
- Construct the symbol table for any input file with the help of LEX tool.
- Consider the context free grammar.  
 $S \rightarrow SS+, S \rightarrow SS*, S \rightarrow a$  and the string  $aa+a^*$ .  
 i) Give the leftmost derivation for the string.  
 ii) Give the rightmost derivation of the string.  
 iii) Is the grammar ambiguous or not.
- Check whether the following grammar is a LL (1) grammar.  
 $S \rightarrow iEtS \mid iEtSeS \mid a, E \rightarrow b$ .
- Construct the FIRST and FOLLOW procedures for the following grammar.  
 $S \rightarrow Aa \mid bAC \mid dc \mid bda, A \rightarrow d$ .
- Consider the grammar,  
 $E \rightarrow TE', E' \rightarrow +TE' \mid \epsilon, T \rightarrow FT', T' \rightarrow *FT' \mid \epsilon, F \rightarrow (E) \mid id$ .  
 Construct a predictive parsing table for the grammar given above. Verify whether the input string  $id + id * id$  is accepted by the grammar or not.

**SKILLS:**

- ✓ Design parsers using top-down and bottom-up approaches.
- ✓ Usage of tools like LEX and YACC.
- ✓ Design a simple code generation

**MODULE-2****UNIT-1****10L+8T+0P=18 hours****INTERMEDIATE REPRESENTATIONS:**

Three address code; Syntax tree; DAG.

**Run-Time Environment:** Storage organization; Stack allocation - Activation Trees, Activation Records.**UNIT-2****14L+8T+0P=22 Hours****OPTIMIZATION AND CODE GENERATION:**

The principal sources of optimization; Basic blocks and flow graphs; Local optimization; Global optimization and loop optimization.

**Code Generation:** Issues in the design of code generator; Code-generation algorithm – register allocation and assignment and peephole optimization.**PRACTICES:**

- Translate the executable statements of the following C-code segment into three address code.  

```
int i;
int a[10]
i = 0;
While (i <= 10) {
a[i] = i + 1; i ++ ;
}
```
- Compute the DAG for the following three address statements. Considering this DAG as an example, explain the process of code generation from DAG.  
 $t1 = a + b$     $t2 = c + d$     $t3 = e - t2$     $t4 = t1 - t3$
- What is Data flow equation? Represent the Data flow information for the following  
 $a = b + c$ ;  $d = c * d$ ;  $e = a - c$ ;  $f = d + e$ .
- Draw a flow graph for the below code. Show the basic blocks clearly in your control flow graph?

```
If (i>=0){
sum = B[0];
i = 0;
L1: if (A[i]< B[i]){
j=i;
L2:
if(B[i]>=0){
sum = sum +B[j];
}
j = j+1
if (j<N) goto L2;
}
i = i+1
if (i<N) goto L1;
}
```

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                       | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply the different phases of compiler with various examples.                                         | Apply        | 1          | 1, 12            |
| 2      | Design different parsing and optimization techniques in the design of compile.                        | Design       | 1          | 1, 2, 12         |
| 3      | Analyze the code optimization techniques.                                                             | Analyze      | 2          | 1, 2, 3, 12      |
| 4      | Analyze the algorithm for compiler segments and evaluate the algorithm for optimized code generation. | Analyze      | 2          | 1, 2, 3,12       |

**TEXT BOOKS:**

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ulman, "Compilers: Principles, Techniques and Tools", 3rd Edition, Pearson Education, 2019.
2. Thomson, "Introduction to Theory of Computation", 2nd Edition, Sipser, 2016.

**REFERENCE BOOKS:**

1. V. Raghavan, "Principles of Compiler Design", 2nd Edition, Mc Graw Hill, 2016.
2. John R. Levin, Tony Mason and Doug Brown, "Lex & YACC", 2nd Edition, O Reilly, 2012.
3. Ms. Manisha Bhamambe, "Compiler Construction", 2nd Edition, Nirali Prakashan, 2017.

# 22AM803 COMPUTER VISION

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Basic Image Processing, Linear Algebra, Vector Calculus, Python Programming with Open CV and Neuroscience.

## COURSE DESCRIPTION AND OBJECTIVES:

This course will cover the fundamentals of Computer Vision. This course provides an understanding of the basic mathematical elements of the image processing and computer vision. It explains and illustrates how the objects are located and identified by the computer in real time environment.

## MODULE-1

### UNIT-1

12L+0T+8P=20 Hours

#### INTRODUCTION TO COMPUTER VISION:

**Basic Concepts Of Image Formation:** Introduction and Goals of Computer Vision and Image Processing, Bio-Vision Vs Computer-Vision, Brief history of Image Formation Concepts.

**Fundamental Concepts Of Image Formation And It's Transformation:** Radiometry, Geometric Transformations, Geometric Camera Models, Camera Calibration, 2D and 3D transformations, 2D and 3D rotations and projections, Image Formation in a Stereo Vision Setup, Image Reconstruction from a Series of Projection.

### UNIT-2

12L+0T+8P=20 Hours

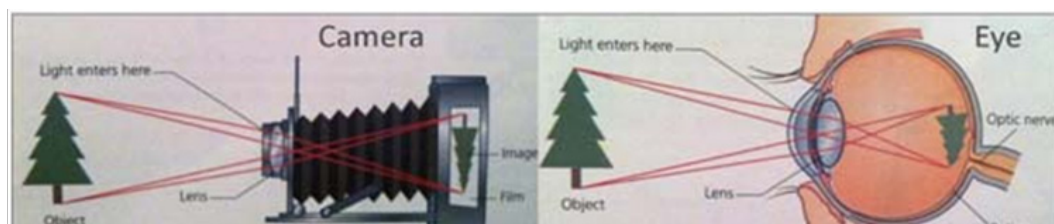
#### IMAGE AND VISION PROCESSING:

**Basic Image Processing:** Image generation, Image Enhancement, Image Filtering, Colour Image Processing, Basic Image Segmentation techniques.

**Image Descriptors and Features:** Texture Descriptors, Colour Features, Edges/Boundaries, Object Boundary and Shape Representations, Interest or Corner Point Detectors, Histogram of Oriented Gradients, Scale Invariant Feature Transform, HoG, LBP, Blobs; Scale Space and Scale Selection; Speeded up Robust Features.

#### PRACTICES:

- Explore and show the practical implementation of the analogy between eye and camera and explain about the vital parts for providing the vision in the human eyes and computers?



Explore and discuss what are the basic image formation techniques and describe with an example how image formation technique got evolved into the present form in terms of the time and space complexities (show the implementations of different algorithms).

Computer vision

Source: <https://www.kdnuggets.com/2020/06/6-easy-steps-implement-computer-vision-application-tensorflow-js.html>

- Implement an intelligent and optimized technique using image arithmetic and logical operations to enhance the given input image. Discuss how to choose the best method depending on the input image.
- Explore an intelligent row wise and column wise algorithm for pixel operations to enhance (highlight the horizontal and vertical edge operations) the given input image.
- Compute the Harris matrix for 3x3 window highlighted below and compare it with 2x2 and 4x4 windows in terms of complexity?

|   |    |    |    |    |
|---|----|----|----|----|
| 0 | 0  | 1  | 4  | 9  |
| 1 | 0  | 5  | 7  | 11 |
| 1 | 4  | 9  | 12 | 16 |
| 3 | 8  | 11 | 14 | 16 |
| 8 | 10 | 15 | 16 | 20 |

| $d/dx$ |   |   |
|--------|---|---|
| -1     | 0 | 1 |

| $d/dy$ |
|--------|
| -1     |
| 0      |
| 1      |

- Discuss and implement a robust technique for performing gray level slicing on medical images with and without background noise.
- Investigate how to revise the technique of histogram equalization to enhance the given input image with salt and pepper noise.
- Write a function to compute the Euclidean distance between the original RGB pixel values and the quantized values. Your function should be called computeQuantizationError, should take in inputs origImg, quantizedImg, and should return an output error, where origImg and quantizedImg are both RGB images, and error is a real number.

## MODULE-2

### UNIT-1

12L+0T+8P=20 Hours

#### VISUAL MATCHING:

**Feature:** Bag-of-words, VLAD, RANSAC, Hough transform, Pyramid Matching, Stereo vision, Correspondence problem, Motion and Optical Flow.

### UNIT-2

12L+0T+8P=20 Hours

#### REAL TIME APPLICATIONS OF COMPUTER VISION:

Artificial Neural Network for Pattern Classification, Gesture Recognition, Motion Estimation, Object detection and Tracking, Face Recognition, Instance recognition, Category recognition, and context and scope understanding.

#### PRACTICES:

- Discuss and compare different image compression techniques in terms of complexity, and propose an intelligent and fast compression technique appropriate to the input image type.
- Given an input image with noise and degradation write an intelligent algorithm to restore the image.
- Your headlights have a radiant intensity of 60 Wsr<sup>-1</sup>. Determine the irradiance on a sign 2 meters away.
- Investigate different existing image (Indoor and outdoor images) segmentation algorithms. Design your own segmentation algorithm on Vadlamudi region (extract image from google map) and compare with the state of the art methods.
- Explore different existing image (Indoor and outdoor images) segmentation algorithms. Design your own segmentation algorithm and compare with the state of the art methods.
- Given an RGB image, perform clustering in the 3-dimensional RGB space, and map each pixel in the input image to its nearest center. That is, replace the RGB value at each pixel with its nearest cluster's average RGB value. Show the output image for different value of K.

#### SKILLS:

- ✓ Acquire the basic image formation knowledge and fundamental image processing techniques.
- ✓ Analyse the various Segmentation techniques for specific applications.
- ✓ Know more advanced topics and current research literature of Image Processing and Computer Vision.
- ✓ Ability to work in industry or in academic research in the field of Computer Vision and Image Processing.

Since these average RGB values may not be integers, you should round them to the nearest integer (1 through 255). Your function should be called `quantize RGB`, should take in inputs `origImg` and `k`, and return outputs `outputImg`, `mean Colors`, `clusterIds`. The variables `origImg` and `outputImg` are RGB images, `k` specifies the number of colors to quantize to, and `mean Colors` is a  $K \times 3$  array of the  $K$  centers (one value for each cluster and each color channel). `clusterIds` is a  $\text{numpixels} \times 1$  matrix (with  $\text{numpixels} = \text{numrows} * \text{numcolumns}$ ) that says which cluster each pixel belongs to.

- A function called `detectEdges` which takes in as input `im`, `threshold` and returns `output edges`. This function computes edges in an image. `im` is the input color image, and `threshold` is a user-set threshold for detecting edges. `edges` is an  $N \times 4$  matrix containing 4 numbers for each of  $N$  detected edge points: the  $x$  location of the point, the  $y$  location of the point, the gradient magnitude at the point, and the gradient orientation (non-quantized) at the point.
  - In this function, first convert the image to grayscale. Then simply compute the gradient magnitude and orientation at each pixel, and only return those  $(x, y)$  locations with magnitude that is higher than the threshold. You can reuse code from HW2.
  - At the end, display, save, and include in your submission the thresholded edge image for image of your choice.
  - Remember that the  $x$  direction corresponds to columns and the  $y$  direction corresponds to rows.

#### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                    | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Explore the fundamental understanding of computer vision and its techniques with its advanced libraries.           | Apply        | 1          | 1, 2, 12         |
| 2      | Implement the techniques to have hands on experience with all the techniques.                                      | Apply        | 1          | 1, 2, 5, 12      |
| 3      | Able to design new changes in methods, and make use of different existing techniques to enhance the vision better. | Apply        | 1          | 1, 2, 3, 5, 12   |
| 4      | Capable to analyse and compare real time models for computer vision problems.                                      | Analyze      | 2          | 1, 2, 3, 12      |
| 5      | Going in depth to find out research outcomes.                                                                      | Analyze      | 2          | 1, 3, 5, 2       |

#### TEXT BOOKS:

1. M.K. Bhuyan, "Computer Vision and Image Processing: Fundamentals and Applications", CRC Press, 2020.
2. Forsyth & Ponce, "Computer Vision-A Modern Approach", 2nd edition, Pearson Education, 2012.

#### REFERENCE BOOKS:

1. R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", 2nd edition, Cambridge University Press, 2004.
2. Simon Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012.
3. Richard Szeliski, "Computer Vision- Algorithms & Applications", Springer, 2010.

# 22AM804 CRYPTOGRAPHY AND NETWORK SECURITY

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Computer networks.

## COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the modern concepts of network security using various cryptographic algorithms and underlying network security applications. It enables to understand various symmetric and asymmetric cryptographic techniques. It focuses on providing security services in applications such as e-mail functioning, web security and secure electronic transactions protocol and system security.

## MODULE-1

### UNIT-1

12L+0T+8P=20 Hours

#### INTRODUCTION:

**Introduction To Computer and Network Security Concepts:** Computer Security Concepts, Security attacks, Security services, Security mechanisms, Fundamental Security Design Principles, Attack Surfaces and Attack trees, A model for network security.

**Classical Encryption Techniques:** Symmetric cipher model, Substitution techniques, Transposition techniques

### UNIT-2

12L+0T+8P=20 Hours

#### SYMMETRIC AND ASYMMETRIC CRYPTOGRAPHY:

**Symmetric Ciphers:** Block cipher principles, Data encryption standard, Strength of DES, Blockcipher design principles, AES cipher, Multiple encryption and triple DES, Block cipher modes of operation, RC4.

**Asymmetric Ciphers and Cryptographic Hash Functions:** Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman Key Exchange, Message Authentication requirements, Authentication functions, Message authentication Codes, Hash functions, Security of hash functions and MACs, Digital signature standard.

#### PRACTICES:

- Implement Substitution and Transposition Ciphers
  - Ceaser cipher
  - Playfair cipher
  - Hill cipher
  - Rail fence cipher
- Implement Symmetric Cipher
  - S-DES
  - RC4
- Implement Asymmetric Cipher
  - RSA
  - Diffie-Hellman
  - Hash Function



Source: [https://www.brainkart.com/subject/CRYPTOGRAPHY-AND-NETWORK-SECURITY-PRINCIPLES-AND-PRACTICE\\_136/](https://www.brainkart.com/subject/CRYPTOGRAPHY-AND-NETWORK-SECURITY-PRINCIPLES-AND-PRACTICE_136/)



**SKILLS:**

- ✓ Design various security services for appropriate applications.
- ✓ Identifying the appropriate firewall, password management and anti-virus models for specific applications.
- ✓ Test and resolve threats and malfunctions in network.
- ✓ Apply different security mechanisms for web applications.
- ✓ Build authentication system for security protocols.

**MODULE-2****UNIT-1****12L+0T+8P=20 Hours****SECURITY APPLICATIONS:****Network Security Applications:** Kerberos, X.509 authentication service, Public key infrastructure,**E-Mail Security:** Pretty good privacy, S/MIME.**IP Security Overview:** IP security architecture, Authentication header, Encapsulating security payload, Combining security associations, key management.**UNIT-2****12L+0T+8P=20 Hours****WEB AND SYSTEM SECURITY:****Web Security:** Secure socket layer and transport layer security, HTTPS, Secure Shell (SSH)**System Security:** Intruders, Intrusion detection, Malicious software, Firewalls**PRACTICES:**

- Configure IP Address in a system in LAN (TCP/IP Configuration)
- Configure DNS to establish interconnection between systems
- Configuring Windows Firewall
- Adding users, setting permissions
- Configure Mail server
- Demonstrate the usage of Wireshark to identify abnormal activity in network communication.
- Demonstrate usage of NMAP (Zenmap) Tool in Network Scanning.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply cryptographic techniques in various security service solutions effectively in everyday professional and social contexts. | Apply        | 1, 2       | 1, 2             |
| 2      | Analyze the usage of secure protocols to safeguard sensitive data using internet.                                              | Analyze      | 1, 2       | 1, 2             |
| 3      | Usage of tools to Identify abnormal activity in network communication to take appropriate action.                              | Apply        | 2          | 5                |
| 4      | Apply various security protocols to safe guard the data internet using SSL/TCL.                                                | Apply        | 2          | 1, 2             |

**TEXT BOOK:**

1. William Stallings, "Cryptography and Network security", 7th Edition, Pearson Education, 2017.

**REFERENCE BOOKS:**

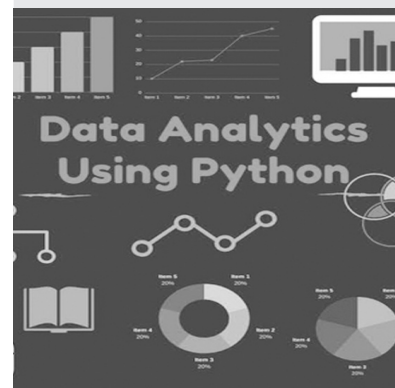
1. William Stallings "Network Security Essentials Applications and Standards", 2nd Edition, Pearson Education, 2009.
2. Eric Malwald, "Fundamentals of Network Security", 4th Edition, Pearson Education, 2010.
3. Buchmann, "Introduction to Cryptography", 2nd Edition, Pearson Education, 2009.
4. Charlie Kaufman, "Radis Perlman and Mike Speciner, Network Security - Private Communication in a Public World", 1st Edition, Pearson Education, 2009.



**22AM805 DATA SCIENCE USING PYTHON**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |



Source: <https://www.fivesquid.com/pics/t2/1622388942-172793-1-1.jpg>

**PREREQUISITE KNOWLEDGE:** Python programming.

**COURSE DESCRIPTION AND OBJECTIVES:**

This course introduces the basic concepts of data science. This course makes students familiar with python libraries to handle various datasets and the methods for cleaning, transforming and enrichment of data. In addition, this course enables the students to train the applications of data science and perform data transformations.

**MODULE-1****UNIT-1****8L+8T+8P=24 Hours****INTRODUCTION TO DATA SCIENCE:**

Steps in doing Data Science - Data Science relation to other fields- Data Science and Information Science- Computational Thinking - Skills and tools needed to do Data Science - Storing data - Combining bytes into larger structures - Creating data sets - Identifying data problem - Understanding data sources - Exploring data models- Introduction to Big Data.

**UNIT-2****8L+8T+8P=24 Hours****DATA ACQUISITION:**

**Import data into Pandas from various data sources:** Fetching stored data- CSV, Excel, Pdf, text, multiple text files, RDBMS (SQL Tables), pickle and JSON; importing data from clipboards, working with binary data formats, Web scraping-beautiful soup for reading and parsing of web pages, reading data from XML, reading data from an API, Reading Image files using PIL, read multiple files using Glob.

**PRACTICES:**

- Write code to perform the following operations:
  - Load the data stored in different files formats: CSV, Excel, txt, ZIP, database, JSON, pickle
  - Saving data into different files formats: CSV, Excel, txt, ZIP, database, JSON, pickle
  - Web Scraping- Use requests module to retrieve data from any website of your interest, improve readability of the extracted data using BeautifulSoup library
  - Reading Image Folders: retrieve images stored in our local folders and visualize few samples from each class
  - Merge multiple data frames, split a data frame into multiple data frames, apply different joins on two data frames
  - Data pre-processing- formatting data to standardize it and make it consistent, normalizing data, grouping data values into bins

**MODULE-2****UNIT-1****8L+8T+8P=24 Hours****DATA CLEANING**

**Sub-Setting, Filtering, And Grouping:** Sub-setting the Data Frame, the unique Function, Conditional Selection and Boolean Filtering, Setting and Resetting the Index, The Group By Method, Aggregating.

**SKILLS:**

- ✓ Data Handling.
- ✓ Data Enriching.
- ✓ Data analysis  
Tool usage.

**Detecting Outliers and Handling Missing Values:** Outlier detection, Missing Values in Pandas, Filling and dropping missing Values in Pandas, Outlier Detection and removing duplicates.

**UNIT-2****8L+8T+8P=24 Hours****DATA ENRICHMENT:**

**Data Integration and transformation:** Combining, merging, and joining data sets, string and text processing using regular expressions, Transforming Numerical features – power transformation, binning, binarization, data transformation based on mappings, Encoding Categorical data- One-hot encoding, Ordinal encoding, Label encoding of the target variable; Scaling- Normalization, Standardization, Robust scaling

**Learning The Hidden Secrets:** Advanced list comprehension and zip function, Date and time data types and tools, Time series basics, Time zone handling.

**PRACTICES:**

- Apply the following operations on the given csv file
  - Load data from CSV files
  - Retrieve first 10, last 10 rows, 3rd Column and a subgroup
  - Query and index operations on the above data frame
  - Insert, delete and update your data
  - Apply aggregate operations
  - Apply various filters on the data
  - Group, merge, and aggregate data in the data frames
- Apply the following operations on the given csv file
  - Load the csv and convert to data Frame
  - Identify the total number of missing values
  - Replace the missing values with a constant, with the Mean of that column, with the mode of that column
  - Remove missing values on the original csv file
  - Apply fill options and replace
- Load a CSV file, change column names, apply required transformations on the data
- Load a CSV file, apply binning to transform numeric data to categorical data
- Load a CSV file, encode the categorical values- transforms non-numerical labels into numerical labels, One-Hot encoding, and Binary encoding
- Merge multiple data frames, split a data frame into multiple data frames, apply different joins on two data frames
- Data pre-processing- formatting data to standardize it and make it consistent, normalizing data, grouping data values into bins
- Work With Datetime Format in Python - Time Series Data, Convert integer to DateTime, Extract DateTime into Different Columns, Assemble DateTime from Different Columns
- Create Pandas DateTime Series using date\_range() method
- Load data that contains dates and times
  - Convert strings to datetime
  - Assemble a datetime from multiple columns
  - Get year, month and day
  - Get the week of year, the day of week, and leap year
  - Get the age from the date of birth
  - Improve performance by setting date column as the index
  - Select data with a specific year and perform aggregation
  - Select data with a specific month and a specific day of the month
  - Select data between two dates
  - Handle missing values

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                  | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply data acquisition tools to collect and visualize                            | Apply        | 1          | 1                |
| 2      | Analyze the data and apply suitable pre-processing and transformation techniques | Analyze      | 1, 2       | 2                |
| 3      | Designing standard datasets for data science projects                            | Develop      | 1, 2       | 3                |
| 4      | Tool usage for data acquisition and engineering                                  | Apply        | 1, 2       | 1, 5             |

**TEXT BOOKS:**

1. Jeffrey S. Saltz, Jeffrey M. Stanton, "An Introduction to Data Science", SAGE Publications, 2018.
2. Wes McKinney, "Python for data analysis", 1st Edition, O'Reilly Media, 2012.

**REFERENCE BOOKS:**

1. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney, 2017.
2. Chirag Shah, "A Hands-On Introduction to Data Science", Cambridge University Press, 2020.

# 22AM806 DATA WAREHOUSING AND DATA MINING



Source: <https://alternative-spaces.com/blog/8-data-mining-techniques-you-must-learn-to-succeed-in-business/>

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Database Management System, Probability and Statistics, Python Programming.

## COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the basic concepts, principles, methods, implementation techniques, Data warehousing and data mining, with a focus on three major data mining functions: (1) Association rule mining (2) Classification and (3) cluster Analysis. It also focuses on issues relating to the feasibility, usefulness, effectiveness and scalability of techniques for the discovery of patterns hidden in large data sets.

## MODULE-1

### UNIT-1

**12L+0T+8P=20 Hours**

#### INTRODUCTION:

**Data Warehouse:** What is Data Warehouse, A Multidimensional Data Model: From tables to data cubes, Stars, snowflake, and fact constellations(schemas for multidimensional databases), Examples for defining star, snowflake, and fact constellation schemas,OLAP operations in the multidimensional data model.

**Data Mining:**What is Data Mining?; Why data mining?; What kinds of data can be mined?;What kinds of patterns can be mined?; Which technologies are used?; What kinds of applications are targeted? Major issues in data mining; Data objects and attribute types; Basic statistical descriptions of data;Measuring Data Similarity and Dissimilarity.

**Data Pre-processing:**Overview - data quality, major tasks in data preprocessing; Data cleaning - missing values, noisy data; Data Integration - entity identification problem, redundancy and correlation analysis tuple duplication; Data value conflict detection and resolution; Data reduction - PCA, attribute subset selection, regression and log linear models; Histogram; Data transformation - data transformation by normalization, Discretization by binning,Histogram Analysis, Concept Hierarchy Generation for Nominal Data.

### UNIT-2

**12L+0T+8P=20 Hours**

#### ASSOCIATION ANALYSIS:

Market basket analysis; Frequent Item sets; Closed item sets and association rules; Frequent Item set Mining Methods-apriori algorithm, generating association rules, improving apriori, FP growth method, vertical format method; Which patterns are interesting? Pattern evaluation method; Pattern Mining in multilevel multidimensional space, Pattern Mining in Multilevel, Multidimensional Space, Constraint based Frequent Pattern Mining.

#### PRACTICES:

- Creation of multidimensional data model and implementation of various schemas using multidimensional data model
- Implement various OLAP operations on multidimensional data models such as SLICE, DICE, ROLLUP, DRILLDOWN and PIVOT
- Apply the following data pre-processing techniques on dataset (download from UCI/ Kaggle/ NCBI data repository) to illustrate the need of the pre-processing in data mining

- a) Data Cleaning
- b) Data Normalization
- c) Data Discretization
- d) Computation of correlation coefficient to analyze the data behavior
- e) Dimensionality reduction using PCA and Wavelets
- Construct Heat Map Table to understand the Correlation among the attributes in a given dataset.
- Extract the interesting association rules from a given dataset using Apriori algorithm.
- Extract the interesting association rules from a given dataset using Frequent Pattern growth algorithm.
- Compare the performance of both filters.
- Develop a module for detecting lines & edges in an image.
- Develop a module for segmenting region of interest.

## MODULE-2

### UNIT-1

**12L+0T+8P=20 Hours**

#### CLASSIFICATION:

What is classification?, General approach to classification, Decision tree induction - attribute selection measures; Tree pruning; Bayes Classification methods - Bayes theorem; Naïve Bayesian classification; Classification by back propagation - a multilayer feed forward neural network; Defining a network topology; Back propagation; K nearest neighbor classifier; Support vector machine, Linearly separable and inseparable cases, Model evaluation and selection; Techniques to improve classification accuracy; Other classification methods – KNN; genetic algorithms; Fuzzy algorithm.

### UNIT-2

**12L+0T+8P=20 Hours**

#### CLUSTER ANALYSIS:

Partition methods - K means and K medoid; Hierarchical methods;

Agglomerative and divisive method; Density based methods – DBSCAN, OPTICS; Optics; Grid based methods-STING, CLIQUE; Cluster evaluation methods; Clustering high dimensional data; Problems, Challenges and major methodologies.

#### PRACTICES:

- Apply the following classifiers on a given dataset and analyze their performance.
  - a) J48 and visualize the decision tree
  - b) Naive Bayes
  - c) Support Vector Machine
  - d) Multi-Layer Perceptron
  - e) K-Nearest Neighbor
- Illustrate the performance of Ensemble Classification algorithms such as Bagging and Boosting Methods
- Evaluate the performance of partitioning clustering algorithms on a given dataset.
- Evaluate the performance of hierarchical clustering algorithms on a given dataset.

#### SKILLS:

- ✓ Handle various types of Data and able to explore the characteristics of data
- ✓ Perform various Data Visualisation tasks over the data and present the data with ease of access
- ✓ Perform descriptive and predictive mining tasks over the data to carry out decision making..

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                      | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Investigate various patterns that can be extracted from different types of data and analyze data warehousing concepts and operations | Analyze      | 1,2        | 1, 2             |
| 2      | Apply various pre-processing techniques and classification algorithms on different domains of data.                                  | Apply        | 1          | 1, 2, 3, 5, 12   |
| 3      | Build decision making systems using data mining algorithms for a given real time data set.                                           | Apply Create | 1, 2, 5, 6 | 1, 2, 3, 5, 12   |
| 4      | Construct models using modern tools such as WEKA, R and Python etc                                                                   | Apply Create | 2          | 1, 2, 3, 5, 12   |

**TEXT BOOKS:**

1. Jiawei Han, Micheline Kamber and Jian Pei, "Data mining Concepts and Techniques", 3rd edition, Morgan Kaufmann. 2012.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", 2nd edition, Pearson, 2018.

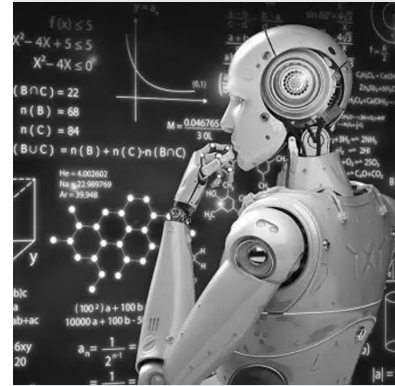
**REFERENCE BOOK:**

1. Jure Leskovec, Anand Raja raman and Jeffrey D Ullman, "Mining of Massive Datasets", 5th edition, Stanford University, 2014.

# 22AM807 NATURE INSPIRED COMPUTING METHODS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 3 |



Source: <https://www.surrey.ac.uk/departments-computer-science/research/nature-inspired-computing-and-engineering-research-group>

**PREREQUISITE KNOWLEDGE:** Data Structures and Algorithms.

## COURSE DESCRIPTION AND OBJECTIVES:

This course covers the fundamentals of nature inspired techniques which influence computing, study the Swarm Intelligence and Immuno computing techniques and familiarize the DNA Computing.

## MODULE-1

### UNIT-1

12L+6T+0P=18 Hours

#### INTRODUCTION, COMPUTING INSPIRED BY NATURE:

**From Nature-to-Nature Computing, Philosophy, Three Branches: A Brief Overview, Individuals, Entities and agents - Parallelism and Distributivity Interactivity.**

**Computing Inspired by Nature:** Evolutionary Computing, Hill Climbing and Simulated Annealing, Darwin's Dangerous Idea, Genetics Principles, Standard Evolutionary Algorithm -Genetic Algorithms.

### UNIT-2

12L+10T+0P=22 Hours

#### SWARM INTELLIGENCE:

Introduction - Ant Colonies, Ant Foraging Behaviour, Ant Colony Optimization, SACO and scope of ACO algorithms, Ant Colony Algorithm (ACA), Swarm Robotics, Foraging for food, Social Adaptation of Knowledge , Particle Swarm Optimization (PSO).

#### PRACTICES:

- How would you implement a GA for a two variable numeric function? What would change: the representation, evaluation function, or genetic operators?
- Write a pseudocode for the simple ACO (S-ACO) algorithm considering pheromone evaporation, implement it computationally, and apply it to solve the TSP instance presented. Discuss the results obtained.
- Apply the AS-TSP algorithm to solve the simple TSP instance. Compare the performance of the AS-TSP with that of the genetic algorithm applied to this same problem. Study and discuss the influence of parameters  $\alpha$  and  $\beta$  in the performance of the algorithm.

## MODULE-2

### UNIT-1

12L+8T+0P=20 Hours

#### IMMUNO COMPUTING:

**Introduction:** Immune System, Physiology and main components, Pattern Recognition and Binding, Immune Network Theory- Danger Theory, Evaluation Interaction Immune Algorithms, Introduction – Genetic algorithms, Bone Marrow Models, Forest's Algorithm, Artificial Immune Networks.

**SKILLS:**

- ✓ Describe the needs for present and future computing paradigm changes with examples found in the real world
- ✓ Able to develop solutions to problems, accompanied by in-depth analysis and insight
- ✓ Demonstrates a competent level of knowledge of NIC concepts, principles, and Applications
- ✓ Design suitable NIC methods to solve problems and model systems in the real-world contexts

**UNIT-2****12L+8T+0P=20 Hours****COMPUTING WITH NEW NATURAL MATERIALS/DNA:**

Computing, DNA Molecule, Adleman's experiment, Test tube programming language, Universal DNA Computers, PAM Model, Splicing Systems, Lipton's Solution to SAT Problem, Scope of DNA Computing, From Classical to DNA Computing.

**PRACTICES:**

- Implement the clonal selection algorithm CLONALG and apply it to the pattern recognition
- Write a test tube program to verify if a given tube  $t$  contains DNA strands with the following sequence of bases: ACTGG.
- A clique  $K_i$  is the complete graph on  $i$  vertices. Given a graph  $G = (V, E)$ , determine the largest  $i$  such that  $K_i$  is a subgraph of  $G$ . This problem is known as the maximum clique problem (MCP). Write a test tube program to solve it, and discuss its functioning.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Understand fundamental concepts of NP-hardness and computational complexity                                    | Apply        | 1          | 1, 2, 12         |
| 2      | Understand the strengths, weaknesses and appropriateness of nature-inspired algorithms.                        | Apply        | 1          | 1, 2, 5, 12      |
| 3      | Apply nature-inspired algorithms to optimization, design and learning problems                                 | Apply        | 1          | 1, 2, 3, 5, 12   |
| 4      | Analyze the Behavior systems of nature inspired algorithm applied in real world problems.                      | Analyze      | 2          | 1, 2, 12         |
| 5      | Understand the theory behind the design of immune networks and DNA computing and their potential applications. | Analyze      | 2          | 1, 2             |

**TEXT BOOK:**

1. Leandro Nunes de Castro, "Fundamentals of Natural Computing, Basic Concepts, Algorithms and Applications", Chapman & Hall/ CRC, Taylor and Francis Group, 2007.

**REFERENCE BOOKS:**

1. Floreano D. and Mattiussi C., "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", MIT Press, Cambridge, MA, 2008.
2. Albert Y.Zomaya, "Handbook of Nature-Inspired and Innovative Computing", Springer, 2006.
3. Marco Dorrigio, Thomas Stutzle, "Ant Colony Optimization", PHI, 2005.



**22AM808 SOFT COMPUTING**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 4 |

**PREREQUISITE KNOWLEDGE:** Design and Analysis of Algorithms.**COURSE DESCRIPTION AND OBJECTIVES:**

Soft computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. Soft computing is based on some biological inspired methodologies such as genetics, evolution, ant's behaviors, particles swarming, human nervous systems, etc. Now, soft computing is the only solution when we don't have any mathematical modeling of problem solving (i.e., algorithm), need a solution to a complex problem in real time, easy to adapt with changed scenario and can be implemented with parallel computing. It has enormous applications in many application areas such as medical diagnosis, computer vision, hand written character recondition, pattern recognition, machine intelligence, weather forecasting, network optimization, VLSI design, etc.

**MODULE-1****UNIT-1****12L+8T+0P=20 Hours****INTRODUCTION:**

Concept of computing systems, "Soft" computing versus "Hard" computing, Characteristics of Soft computing, Some applications of Soft computing.

**UNIT-2****12L+8T+0P=20 Hours****FUZZY LOGIC:**

Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Membership Functions: Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Expert Systems, Fuzzy Decision Making.

**PRACTICES:**

- The membership functions for the linguistic variables "tall" and "short" are given below.

$$\text{tall} = \left\{ \frac{0.2}{5} + \frac{0.3}{7} + \frac{0.7}{9} + \frac{0.9}{11} + \frac{1.0}{12} \right\}$$

$$\text{short} = \left\{ \frac{0.3}{0} + \frac{0}{30} + \frac{1}{60} + \frac{0.5}{90} + \frac{0}{120} \right\}$$

- Develop the membership functions for the following linguistic phrases: (a) Very Tall (b) Fairly tall (c) Not very short .
- Presence a network of fuzzy inference system.

**MODULE-2****UNIT-1****12L+8T+0P=20 Hours****GENETIC ALGORITHMS:**

Goals of optimization, comparison with traditional methods, schemata, Terminology in GA – strings, structure, parameter string, data structures, operators, coding fitness function, algorithm, applications of GA in Machine Learning : Machine Learning Approach to Knowledge Acquisition.



Source: <https://www.hitechnectar.com/blogs/applications-soft-computing/>

**SKILLS:**

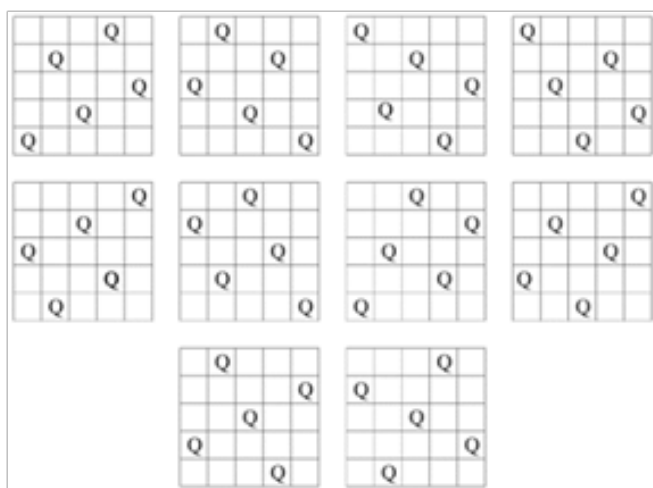
- ✓ Skills to generate Frameworks for fuzzy logic and its application
- ✓ Construct genetic models for hard problems
- ✓ Evaluate and apply.

**UNIT-2****12L+8T+0P=20 Hours****HYBRID SOFT COMPUTING TECHNIQUES AND APPLICATIONS:**

Neuro-fuzzy hybrid systems - genetic neuro hybrid systems - genetic fuzzy hybrid and fuzzy genetic hybrid systems - simplified fuzzy ARTMAP - Applications: A fusion approach of multispectral images with SAR, optimization of traveling salesman problem using genetic algorithm approach, soft computing based hybrid fuzzy controllers.

**PRACTICES:**

- Solve the five queen's problems using Genetic algorithm.



- Solve the TSP problem using Genetic algorithm.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                              | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Analyze real world problems and identify soft computing techniques suitable for solving them | Analyze      | 1          | 1, 2             |
| 2      | Understand soft computing techniques and their roles in building intelligent machines        | Understand   | 1          | 1, 2, 3          |
| 3      | Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems | Apply        | 1          | 1, 2, 3          |
| 4      | Use various tools to solve soft computing problems.                                          | Implement    | 2          | 11, 12           |

**TEXT BOOK:**

1. D. K. Pratihar, "Soft Computing1", Narosa, 2008.

**REFERENCE BOOK:**

1. Kasabov, Nikola K, "Foundations of neural networks, fuzzy systems, and knowledge engineering". Marcel Alencar, 1996.

# 22AM809 INTRODUCTION TO SOFTWARE ENGINEERING

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Data Base Management Systems, OOPS through Java.

## COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the concepts of software life cycle, role of process models and methods to prepare software requirement specification document. In addition to that, it also imparts knowledge of design, development and testing of software. The objective of this course is to enable the student to develop efficient, cost effective, feasible software as per user requirements.

## MODULE-1

### UNIT-1

12L+0T+8P=20 Hours

#### INTRODUCTION TO SOFTWARE ENGINEERING:

Introduction to Software and Software engineering, Evolving role of software, Changing nature of software, Software characteristics, Software project, Software myths, Project Planning, Scheduling and Management.

**Generic View Of Process:** Software Engineering - A layered technology, A process framework, Software Development Life Cycle (SDLC), The Capability Maturity Model Integration (CMMI), Process assessment.

**Process Models:** Conventional Model, Incremental model, Spiral model, Agile process models - Unified process model, Extreme Programming, Scrum.

### UNIT-2

12L+0T+8P=20 Hours

#### REQUIREMENTS ENGINEERING:

Functional and Non-functional requirements, User requirements, System requirements, Requirement engineering tasks, formal requirements specification and verification, Feasibility Study.

**Building The Analysis Model:** Data modeling - Data objects, Attributes, Relationships, Cardinality and modality. Class based modeling - Identify analysis classes, specify attributes and Define operations.

**Design Engineering:** Design model, Design concepts. Creating an Architectural Design-Architectural styles and patterns.

**Performing User Interface Design:** Golden rules; User interface analysis and design.

#### PRACTICES:

Laboratory session of this course is designed in such a way that the student should complete three projects of the given type by performing the below experiments.

- Development of software requirements specification using Mind-Map tool.
- Project planning using Gantt charts.
- Project estimation using metrics.
- Capture Use Case Scenarios and model UML Use Case Diagrams.
- Model the UML state chart and Activity diagrams.
- Model the UML Class and Sequence diagrams.

## Software Engineering



Source: <https://artoftesting.com/software-engineering>

**SKILLS:**

- ✓ Know the software requirements and find out various ways to gather and specify them.
- ✓ Choose a process model for developing software solutions without schedule/ effort overruns and good quality.
- ✓ Analyse and model (diagrammatical representations) a software product.

**MODULE-2****UNIT-1****12L+0T+8P=20 Hours****TESTING STRATEGIES:**

A strategic approach to software testing, Unit testing, Integration testing, Validation testing, System testing, Art of debugging

**Testing Tactics:**Black-Box testing- Graph based testing, Equivalence partitioning, Boundary value analysis, White-Box testing – basis path testing, Control structure testing.

**Product Metrics:**Metrics for analysis model; Metrics for design model, Metrics for source code; Metrics for testing; Metrics for maintenance.

**UNIT-2****12L+0T+8P=20 Hours****RISK MANAGEMENT:**

Software risks, Risk identification; Risk projection; Risk refinement, Reactive vs Proactive risk strategies, RMMM, RMMM plan.

**Quality Management:** Quality concepts, Software quality assurance, Software reviews, Statistical Software Quality Assurance, ISO 9000 quality standards.

**Computer-Aided Software Engineering (CASE):** Use of appropriate CASE tools- Requirement engineering tools, Project planning tools, Testing tools.

**PRACTICES:**

Laboratory session of this course is designed in such a way that the student should complete three projects of the given type by performing the below experiments.

- Estimate the test coverage and Structural complexity of product using metrics.
- Develop the test cases for all the functional requirements of projects selected.
- Perform the functional testing using Selenium tool.

**LIST OF PROJECTS:**

**Project-1:** A Point-Of-Sale (PoS) System: A POS system is a computerized application used to record sales and handle payments; it is typically used in a retail store, it includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services are temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client-side terminals and interfaces such as browser, PDAs, touch-screens.

**Project-2:** Online Bookshop Example: Following the model of amazon.com or bn.com, design and implement an online bookstore.

**Project-3:** A Simulated Company: Simulate a small manufacturing company. The resulting application will enable the user to take out a loan, purchase a machine, and over a series of monthly production runs, follow the performance of their company.

**Project-4:** A Multi-Threaded Airport Simulation: Simulate the operations in an airport. Your application should support multiple aircrafts using several runways and gates avoiding collisions/ conflicts. Landing: an aircraft uses the runway, lands, and then taxis over to the terminal. Take-Off: an aircraft taxis to the runway and then takes off.

**Project-5:** An Automated Community Portal: Business in the 21st Century is above all BUSY. Distractions are everywhere. The current crop of “enterprise intranet portals” is often high noise and low value, despite the large capital expenditures it takes to stand them up. Email takes up 30 - 70% of an employee’s time. Chat and Instant Messaging are either in the enterprise or just around the corner. Meanwhile, management is tasked with unforeseen and unfunded leadership and change-agent roles as well as leadership development and succession management. What is needed is a simplified, repeatable process

that enhances communications within an enterprise, while allowing management and peers self-select future leaders and easily recognize high performance team members in a dynamic way. Additionally, the system should function as a general-purpose content management, business intelligence and peer-review application. Glasscode's goal is to build that system.

**Project-6:** Content Management System: The goal is to enable non-technical end users to easily publish, access, and share information over the web, while giving administrators and managers complete control over the presentation, style, security, and permissions. Features: Robust Permissions System, Templates for easy custom site designs, Total control over the content, Search engine friendly URL's, Role based publishing system, Versioning control, Visitor profiling.

**Project-7:** An Auction Application: Several commerce models exist and are the basis for several companies like eBay.com, priceline.com etc. Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.

**Project-8:** A Notes And File Management System: During one's student years and professional career one produces a lot of personal notes, documents. All these documents are usually kept on papers or individual files on the computer. Either way the bulk of the information is often to be erased, corrupted and eventually lost. The goal of this project is to build a distributed software application that addresses this problem. The system will provide an interface to create, organize and manage personal notes through the Internet for multiple users. The system will also allow users to collaborate by assigning permissions for multiple users to view and edit notes.

**Project-9:** Library Management System(LMS): The goal is to enable students and librarians to easily access and manage the library and run it smoothly. Each physical library item - book, tape cassette, CD, DVD, etc. could have its own item number. To support it, the items may be barcoded. The purpose of barcoding is to provide a unique and scannable identifier that links the barcoded physical item to the electronic record in the catalog. Barcode must be physically attached to the item, and barcode number is entered into the corresponding field in the electronic item record. Barcodes on library items could be replaced by RFID tags. The RFID tag can contain item's identifier, title, material type, etc. It is read by an RFID reader, without the need to open a book cover or CD/DVD case to scan it with barcode reader.

**Project-10:** Hospital Management System: Simulate to show and explain hospital structure, staff, and relationships with patients, and patient treatment terminology

**Project-11:** Draft Software Requirement Analysis for the following Problem Statement: Fuel Delivery System : An unattended petrol (gas) pump system that includes a credit card reader. The customer swipes the card through the reader and then specifies the amount of fuel required. The fuel is delivered, and the customer's account debited.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                     | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Use basic concepts of software engineering for designing software product                                                           | Usage        | 1          | 1, 11            |
| 2      | Compare different process models and identify appropriate process model based on project requirements                               | Evaluation   | 1          | 2, 4             |
| 3      | Build Software Requirement Specification (SRS) document for any software product                                                    | Design       | 1          | 3, 5             |
| 4      | Design of solutions using UML diagrams like Usecase, Sequence diagrams etc                                                          | Design       | 1          | 3, 4, 5          |
| 5      | Create an appropriate architecture for a given project that meets all quality constraints                                           | Create       | 2          | 5                |
| 6      | Apply different testing techniques to ensure bug free software and metrics to measure the software size, complexity, and budget etc | Apply        | 2          | 4, 5, 11         |

**TEXT BOOKS:**

1. Roger S. Pressman, "Software Engineering, A practitioner's Approach", 6th edition, McGrawHill International edition, 2008.
2. Booch G., Rumbaugh J. and Jacobsons I, "The Unified Modeling Language User Guide", 2nd edition, Addison Wesley, 2005.

**REFERENCE BOOKS:**

1. Simon Sennet, Steve McRobb and Ray Farmer, "Object Oriented Systems Analysis and Design, 2nd edition, 2004.
2. Dr. Pankaj Jalote "Software Engineering: A Precise Approach" –edition 2010.

# 22AM810 INTERNET OF THINGS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |



Source: <https://toolsense.io/glossary/iot/>

**PREREQUISITE KNOWLEDGE:** Computer networks.

## COURSE DESCRIPTION AND OBJECTIVES:

Internet of Things (IoT) is presently a hot technology worldwide. Government, academia, and industry are involved in different aspects of research, implementation, and business with IoT. IoT cuts across different application domain verticals ranging from civilian to defense sectors. These domains include agriculture, space, healthcare, manufacturing, construction, water, and mining, which are presently transitioning their legacy infrastructure to support a IoT. To introduce the terminology, technology and its applications, to introduce the concept of M2M (machine to machine) with necessary protocols.

## MODULE-1

### UNIT-1

**12L+0T+8P=20 Hours**

#### INTERNET OF THINGS FUNDAMENTALS:

Introduction to Internet of Things; Physical design & Functional Block of IoT, Device architectures, Core IoT Functional Stack; Resource constrained devices; Sensors and Components; IoT Enabling Technologies. Societal Benefits of IoT (Domain Specific), Risks, Privacy, and Security.

**Network And Communication Protocols:** Network Components; Internet Structure, Wireless Protocols; IoT Communication Model & APIs, Wireless Protocol Stack, IoT levels.

### UNIT-2

**12L+0T+8P=20 Hours**

#### IOT AND M2M:

Software defined networks, Network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCONF-YANG: SDN; NFV; Need for IOT Systems Management; SNMP-NETCONF, YANG; IOT Systems management with NETCONF-YANG.

#### PRACTICES:

- Identify different Sensors and IoT devices
- Identify the Components in Raspberry pi, Arduino, and UNO boards
- Examine IoT levels with any one domain specific application like home automation, weather monitoring system etc.
- Design the Network Configuration and System Management with IoT devices using NETCONF-YANG.
- Design the Network Configuration and System Management with IoT devices using SNMP-NETCONF.

## MODULE-2

### UNIT-1

**12L+0T+8P=20 Hours**

#### INTRODUCTION TO SYSTEMS DESIGN & DEVELOPMENT:

IoT system building blocks, Arduino, Node MCU– Board details, IDE programming; Raspberry Pi-Model and Interfaces, Platform: Axonize, Blynk IoT platform, Fogwing.

**SKILLS:**

- ✓ *Sensor Identification and IoT system design.*
- ✓ *Sensor data analysis.*
- ✓ *Tool usage for developing IoT applications.*

**UNIT-2****12L+0T+8P=20 Hours****PROGRAMMING AND CASE STUDY:**

Embedded C vs Python; Operating systems for constrained devices; Domain Specific IoT Application, Task Support IoT Example: The Refrigerator, Weather Monitoring System – Case study- Design, Programming and Execution.

**PRACTICES:**

- Demonstration and study of Raspberry Pi board, GPIO Pins and familiarity of various sensors.
- Demonstration and study of other Hardware board of IoT such as Arduino Uno and NodeMCU.
- Design and Implementation of controlling LED-using Python in Raspberry Pi board.
- Design and Implementation of sensing light through LDR using Python in Raspberry Pi board.
- Design and Implementation to find obstacles through sensor using Python in Raspberry Pi board.
- Design and Implementation of sensing and display temperature using Python in Raspberry Pi board.
- Design and Implementation of detecting noise through microphone sensor using Python in Raspberry Pi board.
- Design and Implementation of output devices through relay using Python in Raspberry Pi board.
- Design and Implementation of vibration sensor using Python in Raspberry Pi board.
- Design and Implementation of uploading sensor data into cloud using Python.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                      | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Design an end-to-end Machine-learning model to realize solutions for real-world problems.            | Design       | 1          | 3                |
| 2      | Apply various machine-learning models to develop IoT applications.                                   | Apply        | 2          | 1                |
| 3      | Illustrate the impact and challenges posed by IoT networks leading to new architectural models.      | Analyze      | 1          | 4, 6             |
| 4      | Compare and contrast the deployment of smart objects and the technologies to connect them to network | Evaluate     | 2          | 4                |

**TEXT BOOK:**

1. Arsh deep Bahga and Vijay Madiseti "Internet of Things: A Hands-on Approach", Universities Press, 2015, ISBN: 9788173719547.
2. Rajkumar Buyya and Amir Vahid Dastjerdi "Internet of Things: Principles and Paradigms", Morgan Kaufmann; 1st edition, May 25, 2016.

**REFERENCE BOOKS:**

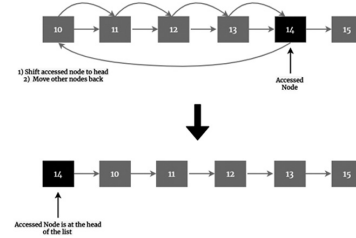
1. Matt Richardson & Shawn Wallace "Getting Started with Raspberry Pi", O'Reilly (SPD), 2014, ISBN: 9789350239759.
2. Walteneus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice".
3. Beginning Sensor networks with Arduino and Raspberry Pi – Charles Bell, A press, and 2013.



# 22CS801 ADVANCED DATA STRUCTURES

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |



**PREREQUISITE KNOWLEDGE:** Data structures.

## COURSE DESCRIPTION AND OBJECTIVES:

Advanced data structures are one of the most important disciplines of data science since they are used for storing, organizing, and managing data and information to make it more efficient, easier to access, and modify. The objective of this course is to improve students' skills in designing data structures and algorithms for various solutions, and problem design in large systems and applications such as databases, information retrieval systems, bioinformatics, and geographic information systems.

## MODULE-1

### UNIT-1

8L+8T+8P=24 Hours

#### NUMBER THEORY & COMPUTATIONAL GEOMETRY:

**Number Theory:** Motivation and concept of modulo, Arithmetic operations under modulo, Euclidean algorithm, extended Euclidean algorithm.

**Computational Geometry:** One Dimensional Range Searching, Two-Dimensional Range Searching.

### UNIT-2

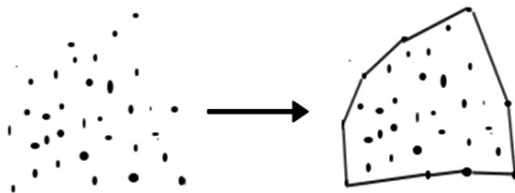
8L+8T+8P=24 Hours

#### TREES:

**Trees:** Splay tree, Red Black Tree (RBT), Operations on RBT, M-way search tree and operations, Segment tree.

#### PRACTICES:

- Find GCD of a given number using Euclidean and extended Euclidean algorithms.
- Check whether a given number is prime or not using Sieve of Eratosthenes.
- Implement a program to find the closest pair of points using a divide and conquer strategy. Use the random number generator to generate a large number of points in a unit square as input to the algorithm. Test the correctness of the algorithm by using a brute force method.
- Given a set of points in the plane. the convex hull of the set is the smallest convex polygon that contains all its points of it.



- Implementation of a Red-Black tree operation.
- Implementation of a Splay tree operation.
- You are given a tree with  $n$  vertices (numbered  $1, 2, \dots, n$ ) and an integer  $k$ .
- A subtree is defined as a connected subgraph of the tree. That is, a subtree is another tree that can be obtained by removing some (possibly none) vertices and all edges incident to those vertices from  $T$ .

Source: <https://www.baeldung.com/cs/advanced-data-structures>

**SKILLS:**

- ✓ Analyse the data structure required for various applications.
- ✓ Usage of trees, graphs, heaps and tries.
- ✓ Know various pattern searching algorithms.

A subset S of vertices is called good if every subtree containing all the nodes in S has at least k nodes.

Find the size of the smallest good subset.

**Input:**

The first line contains a single integer T, the number of test cases. The descriptions of test cases follow.

The first line of each test case contains two integers, n, and k.

The next n-1 lines each contain two integers u, v, denoting an edge between vertices u and v of the tree.

**Output:**

For each test case print in a single line, the minimum size of a good subset.

**MODULE-2****UNIT-1****8L+8T+8P=24 Hours****QUEUES & HEAPS:**

**Priority:** Priority Queues: Introduction, Types of priority queues, implementing max priority queue and min priority queue.

**Heaps:** Introduction, types of heaps, heap implementation with an array, priority queue implementation with heaps, heap sort, technique to sort elements, Mergeable heaps, Fibonacci Heaps and operation.

**UNIT-2****8L+8T+8P=24 Hours****PATTERN SEARCHING & TRIES:**

**Pattern Searching:** Pattern matching algorithms -Brute force, the Boyer —Moore algorithm, Robin-Karp algorithm, Knuth-Morris-Pratt algorithm.

**Tries:** Standard Tries, Compressed Tries, Suffix tries.

**PRACTICES:**

- Implement different operations on Priority Queue. i.e. adding an element, removing an element, size of the priority queue, printing the queue, and top element of the queue.
- You are given N numbers you can decrease each number by 10 % or K kg whichever is more. If the number, you choose is less than K you can make it zero. Your task is to minimize the sum of the number as small as possible by performing N operations only. [Note: Use Priority Queue].

**Example:****Input:**

N = 2

k = 10

arr = 100 15

**Output:**

95

**Explanation:**

You will remove 10 from 100

then once again remove 10 90.

- Given a Binary Heap of size N in an array arr[]. Write a program to calculate the height of the Heap.

**Input:** N = 6

arr = {1, 3, 6, 5, 9, 8}

**Output:** 2

- Given two binary max heaps as arrays, merge the given heaps to form a new max heap.

**Example:**

**Input :**

$n = 4$   $m = 3$

$a[] = \{10, 5, 6, 2\}$ ,

$b[] = \{12, 7, 9\}$

**Output :**

$\{12, 10, 9, 2, 5, 7, 6\}$

- In Doraland, people have unique Identity Numbers called D-id. Doraemon owns the most popular gadget shop in Doraland. Since his gadgets are in high demand and he has only  $K$  gadgets left he has decided to sell his gadgets to his most frequent customers only.  $N$  customers visit his shop and D-id of each customer is given in an array  $array[]$ . In case two or more people have visited his shop the same number of time he gives priority to the customer with higher D-id. Find the D-ids of people he sells his  $K$  gadgets to.

Example:

**Input:**

$N = 6$

$array[] = \{1, 1, 1, 2, 2, 3\}$

$K = 2$

**Output:**

1 2

**Explanation:**

Customers with D-id 1 and 2 are most frequent.

- You are given  $q$  queries of two types:

X: Append value  $X$  into an array.

X K: You are required to print the  $K$ th minimum XOR of  $X$  with the current array.

You have to make a new array whose  $i$ th element is  $current\_array[i] \oplus X$ . Then sort it and print the  $K$ th element.

**Input format**

The first line contains  $q$  ( $1 \leq q \leq 100000$ ).

Next  $q$  lines contain the types of queries. (1 or 2) If type is 1, then it contains  $X$  ( $1 \leq X \leq 10^8$ ).

If type is 2, then it contains  $X$  ( $1 \leq X \leq 10^8$ ) and  $K$  ( $K \leq$  current array size).

**Output format**

Print the number in the second type of query. (Note: Use Tries)

- Given an array of strings  $arr[]$  of size  $n$  and given  $s$  a string  $str$  and an integer  $k$ . The task is to find the count of strings in  $arr[]$  whose prefix of length  $k$  matches with the  $k$  length prefix of  $str$ . [Note: Use Tries]

**Input:**

$n = 6$

$arr[] = \{“abba”, “abbb”, “abbc”, “abbd”, “abaa”, “abca”\}$

$str = “abbg”$

$k = 3$

**Output:** 4

**Explanation:**

“abba”, “abbb”, “abbc” and “abbd” are the matching strings.

- Given an array of integers of size N find minimum xor of any 2 elements. [Note: Use Tries]

**Input:**

N = 3

arr[] = {9,5,3}

**Output:** 6**Explanation:**

There are 3 pairs –

 $9^5 = 12$  $5^3 = 6$  $9^3 = 10$ 

Therefore output is 6.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                         | Blooms Level | Module No. | Mapping with POs |
|--------|-------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply logic and methods behind the major proofs in Number Theory.                                                       | Apply        | 1          | 1, 2             |
| 2      | Implement different types of trees and apply them to problem solutions.                                                 | Apply        | 1          | 1, 2, 3          |
| 3      | Apply the concepts of advanced Trees for solving problems effectively.                                                  | Apply        | 1          | 1, 2, 3          |
| 4      | Analyze how efficient pattern matching principles can be used in the design and implementation of qualitative research. | Analyse      | 2          | 1, 2, 3, 4       |
| 5      | Analyze the given scenario and choose the appropriate Data Structure for solving problems.                              | Analyse      | 2          | 1, 2, 3, 4       |

**TEXT BOOKS:**

- Karl-Dieter Crisman, Number Theory: In context and Interactive by AIM Open Textbook Initiative 7th Edition, 2021.
- Sahani, Anderson freed, Horowitz, Fundamentals of DATA STRUCTURES in C, Silicon Pr; 2nd Edition, August 1, 2007.

**REFERENCE BOOKS:**

- Mark de Berg, Otfried Cheong, Marc van Kreveld, Computational Geometry: Algorithms and Applications:, Springer; 3rd Edition. 2008.
- Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004
- T.Cormen, R.Rivest, C.Stein, C.Leiserson, "Introduction to Algorithms", , PHI publication, Second Edition, 2004, ISBN 81-203-2141-3.

**22CS802 ADVANCED JAVA PROGRAMMING**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |



Source: <https://www.kpl.gov/catalog/item/?i=ent://LYNDA/0/LYNDA:103608>

**PREREQUISITE KNOWLEDGE:** Programming JAVA.

**COURSE DESCRIPTION AND OBJECTIVES:**

Advanced Java programming covers the standard concepts such as database connectivity, web-services, web application development etc. It is specially designed to develop web-based, network-centric or enterprise applications. It simplifies the complexity of building an n-tier application.

**MODULE-1****UNIT-1****8L+8T+8P=24 Hours****INTRODUCTION TO JAVA WEB FRAMEWORKS:**

**Introduction to Java Web frameworks:** What is Framework in Java, Advantages of Frameworks, Popular Java Web Frameworks: Spring, Spring Boot, Hibernate, Java Server Faces, Google Web Toolkit etc., Comparison of the web frameworks. Environmental setup to use Web Frameworks.

**UNIT-2****8L+8T+8P=24 Hours****ORM AND HIBERNATE:**

**ORM:** What is Object Relational Mapping, How ORM Works, Features of ORM, Advantages, Java ORM- Hibernate, JAVA Persistence API(JPA), ORM implementation.

**Hibernate:** Overview of Hibernate, Hibernate Architecture, Hibernate Mapping Types, Hibernate O/R Mapping, Hibernate Annotation, Hibernate Query Language.

**PRACTICES:**

- Installation and Environmental setup to work with java Web Frameworks.
- Create a simple application of hibernate using XML.
- Create a maven based hibernate application using annotation
- Develop a web application using hibernate framework.
- Using the web application to perform database access with hibernate framework.

**MODULE-2****UNIT-1****6L+6T+6P=18 Hours****STRUTS:**

**Struts:** Struts framework, Struts features, Model 1vs Model 2 (MVC) Architecture, Core components of Struts, Configuration of Struts, Creating a Struts Application.

**UNIT-2****10L+10T+10P=30 Hours****SPRING AND SPRING BOOT:**

**Spring:** Overview of Spring, Spring Architecture, Inversion of Control (IOC) and Dependency Injection, XML Configuration on Spring, Creating a Spring Application. Spring MVC, flow of Spring Web MVC, Spring Web MVC Framework Example.

**Spring Boot:** Overview of Spring Boot, Spring vs Spring MVC vs Spring Boot, SB architecture, Components of SB, Creating Spring Boot Application.

**SKILLS:**

- ✓ To know the importance of java web frameworks to develop enterprise java applications.
- ✓ To experience developing a web application using MVC architecture.
- ✓ Design and develop CRUD applications using different java frameworks.
- ✓ To derive hands on experience of developing enterprise applications using advanced web frameworks like Spring and Spring Boot.

**PRACTICES:**

- Develop a simple web application using struts.
- Develop a CRUD (Create Read Update Delete) Application using spring.
- Implement Spring MVC Pagination, which is used to display a large number of records in different parts. Display 10, 20 or 50 records in one page.
- Using Spring MVC upload a file.
- Create a self-contained HTTP application that uses embedded server like Tomcat using Spring Boot.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                          | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Implement web frameworks to develop web applications using java.                         | Apply        | 1          | 1, 2, 3          |
| 2      | Design java web applications using MVC architecture using web frameworks like hibernate. | Create       | 1          | 1, 2, 3          |
| 3      | Implement java application to interact with database using struts.                       | Apply        | 2          | 1, 2, 3          |
| 4      | Design secure transaction-based web applications using Spring framework.                 | Create       | 2          | 1, 2, 3, 4       |

**TEXT BOOKS:**

1. James Keogh, "Complete Reference J2EE", 9th Edition, McGraw Hill, Education, 2002.
2. Jeff Linwood and Dave Minter, "Beginning Hibernate", 2nd Edition, Après publication 2010.

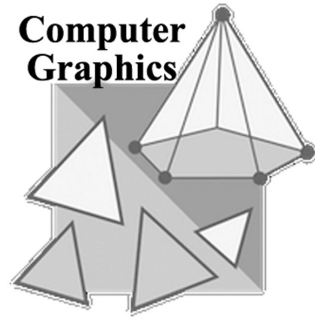
**REFERENCES BOOKS:**

1. Sharanam Shah, Vaishali Shah. "Struts 2 for Beginners", 3rd Edition, Arizona Business Alliance, 2014.
2. Craig walls, "Spring in Action", 5th Edition, Manning Publication, 2018.
3. <https://struts.apache.org/getting-started/index.html>.
4. <https://www.javatpoint.com/hibernate-tutorial>.
5. <https://www.dineshonjava.com/spring-tutorial/>.

**22CS803 COMPUTER GRAPHICS**

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 2 | 2 | 4 |



Source: <https://www.javatpoint.com/computer-graphics-tutorial>

**PREREQUISITE KNOWLEDGE:** Basic Logical Thinking and Problem Solving Ability.

**COURSE DESCRIPTION AND OBJECTIVES:**

This course provides a comprehensive introduction to computer graphics leading to understanding the contemporary terminology and algorithms of computer graphics. To make the students learn the basic principles of visualization. To give an introduction to 2D and 3D modelling and animation.

**MODULE-1****UNIT-1****8L+8T+8P=24 Hours****OVERVIEW OF GRAPHICS SYSTEM INTRODUCTION:**

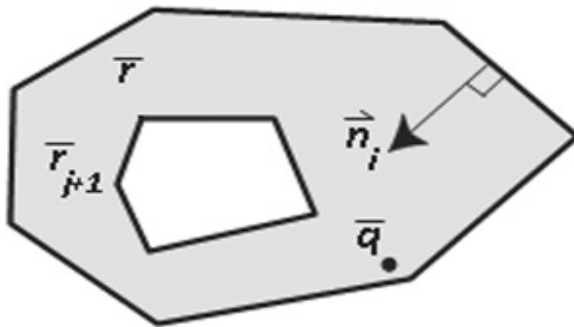
Application areas of Computer Graphics; Video-display devices: raster-scan systems and random scan systems; Graphics primitives: display devices, primitive devices; Filled area primitives: scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

**UNIT-2****8L+8T+8P=24 Hours****2D GEOMETRIC TRANSFORMATIONS AND VIEWING BASIC TRANSFORMATION:**

Geometric Transformations, matrix representations, and homogeneous coordinates. Inverse transformations, 2D Composite transformations, other 2D transformations, raster methods for geometric transformations; 2-D Viewing: The viewing pipeline, window to viewport coordinate transformation, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

**PRACTICES:**

- **Points & Polygons:** Suppose the vertices of a convex polygon are  $p^{-1}, \dots, p^{-N}$ , given in counter-clockwise order. Let the coordinates of vertex  $p^{-i}$  be  $(x_i, y_i)$ .



Describe an algorithm that can tell whether a 2D point  $q^{-}$  is inside the gray-shaded region in the figure. You should assume that the polygons defined by vertices  $p^{-1}, \dots, p^{-N}$  and  $r^{-1}, \dots, r^{-M}$ , respectively, are both convex. Hint: Each edge is contained in an infinite line. Each infinite line divides the 2D plane into two half-planes, the left half-plane and the right half-plane (here left/right means left/right with respect to a counter-clockwise direction of traversal of the vertices). The key insight you should use is that the interior of a convex polygon is the intersection of the left half-planes of all the edges of the polygon.

**SKILLS:**

- ✓ Learn various clipping algorithms.
- ✓ Compare 2-D and 3\_D transformations.
- ✓ Learn various surface detection methods

- Transformations & Commutativity: We say that the 2D transformations  $f()$  and  $g()$  commute if and only if  $f(g(\bar{p})) = g(f(\bar{p}))$  for all points  $\bar{p} \in R^2$ . For each of the four cases below, where  $f()$  and  $g()$  are homographies, prove whether or not they commute:
  - a) Both  $f()$  and  $g()$  are arbitrary homographies.
  - b) One is an arbitrary rotation and the other an arbitrary translation.
  - c) One is an arbitrary translation and the other is a non-uniform scaling.
  - d) One is an arbitrary rotation and the other is a reflection. In each case, your solution can either be a derivation that proves/disproves commutativity or, if  $f()$  and  $g()$  do not commute, a specific counter-example.
- Contrast the implementation of Display File/Frame Buffer for a Random Scan System & Raster Scan System. Which type of system shall offer more consistent refresh rate and why?
- Determine the most appropriate pixels that will be plotted when Bresenham's algorithm is used to draw a line joining the points (10,20) and (20,30)
- Suppose a system with 8 inches by 10 inches video monitor that can display 100 pixels per inch. If memory is organized as one byte words, the starting frame buffer address is 0 and each pixel is assigned one byte of storage in memory, what is the frame buffer address of pixel with screen coordinates (x,y) ? Also, determine the total amount of memory consumed by the frame buffer.
- Consider a raster monitor of resolution 640\*480 pixels. A scanning is used with horizontal retrace time of 4 micro seconds and vertical retrace time of 20 micro seconds respectively. Calculate the time available to display a pixel for both cases of (i) non-interlaced and (ii) interlaced. Assume a scan rate of 50 frames.

**MODULE-2****UNIT-1****8L+8T+8P=24 Hours****3D GEOMETRIC TRANSFORMATIONS:**

3-D Object representation, 3-D geometry primitives, transformations, projection clipping. Interaction: Hardware input devices handling algorithms, Event handling echoing, Interactive techniques, 3-D Geometric transformations, Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3D viewing: Viewing pipeline, viewing coordinates, general projection transformation and clipping.

**UNIT-2****8L+8T+8P=24 Hours****VISIBLE SURFACE DETECTION METHODS:**

Back-face detection, depth buffer, A-buffer, Z-buffer, scan-line Illumination Models, and Surface rendering Methods: Basic illumination models, polygon rendering method.

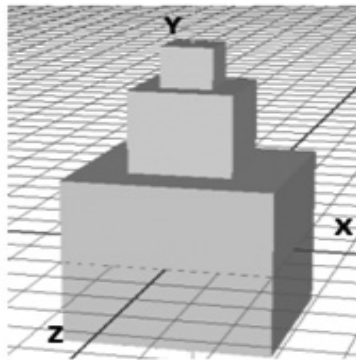
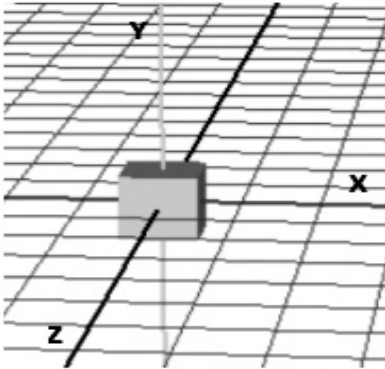
**PRACTICES:**

- Matrix implementation is a powerful technique used for computer graphics. For the concerns stated in section II, matrix for 3D space is represented as a 4X4 array instead of 3X3 array.
- Each operation of object such as translation and rotation can be represented by one Matrix. By multiplying the matrix, a vector/point can be transformed by the operation. For example,  $P = (x, y, z, 1)$  is a point's coordinator before the operation, and  $M = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \end{pmatrix}$  is the matrix representation of translating the object a blocks right, b blocks up and c blocks forward, then after the translating, the point's coordinator will become  $P' = P * M$  is the coordinator of the point after the operation.
- A sequence of operation can also be composed to be one operation. For example, matrix  $M_1$  represent first operation, matrix  $M_2$  represent second operation, the matrix  $M = M_1 * M_2$  will represent this sequence of 2 operations. If  $P = (x, y, z, 1)$  is the coordinator of the point before perform the sequence of the operations, then  $P' = P * M$  will be the coordinator of the point after the perform the operation1 then perform the operation2.



Following are matrix representation for some operations:

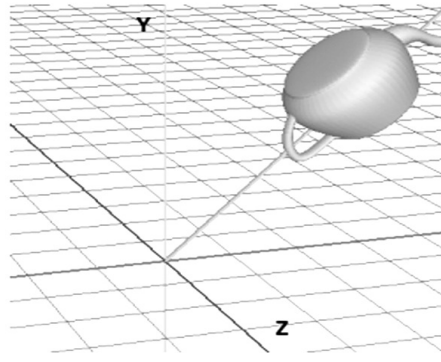
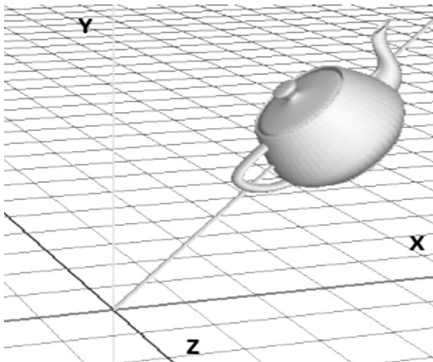
- Translate object by  $V$ , where  $V$  is the vector:  
 $M = \begin{pmatrix} 1 & 0 & 0 & V[0] \\ 0 & 1 & 0 & V[1] \\ 0 & 0 & 1 & V[2] \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Rotate object along X axis by  $d$  degree:  
 $M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(d) & -\sin(d) & 0 \\ 0 & \sin(d) & \cos(d) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Rotate object along Y axis by  $d$  degree:  
 $M = \begin{pmatrix} \cos(d) & 0 & \sin(d) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(d) & 0 & \cos(d) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Rotate object along Z axis by  $d$  degree:  
 $M = \begin{pmatrix} \cos(d) & -\sin(d) & 0 & 0 \\ \sin(d) & \cos(d) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Start with a unit cube centered at the origin. Use two more cubes like this one and obtain the model shown in the right image, where each new cube is twice the size of the previous one



Below list the transformations in the order in which they were applied

1) scaling    2) Translation    3) Rotation

- There are situations in which rotations about axes other than the three coordinate axes are useful. In the next two activities, you will discover how a rotation about an arbitrary axis can be broken down into a series of rotations about the three coordinate axes.
- The yellow axis passing through the teapot in the image below is parallel to the vector  $(1, 1, 0)$ . Determine a sequence of coordinate axis rotations that result in the teapot rotating 180 degrees about the yellow axis, as shown in the right image.



Write the sequence of rotations in the order in which you applied them. Hint: use a rotation that aligns the yellow axis with the x-axis, then rotate the object about the x-axis, then use the reverse rotation that brings the yellow axis back to original position.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|----------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Implement of various graphics systems along with the algorithms used for these devices for drawing.            | Apply        | 1          | 1, 2             |
| 2      | Demonstrate 2D graphics and algorithms including line drawing, polygon filling, clipping, and transformations. | Apply        | 1          | 1, 2             |
| 3      | Demonstrate Concepts and techniques used in 3D computer graphics and basic about animation.                    | Apply        | 2          | 1, 2, 3          |
| 4      | Implement Surface detection methods to get a realistic screen image.                                           | Apply        | 2          | 1, 2,3           |

**TEXT BOOKS:**

1. Donald Hearn and M. Pauline Baker, "Computer Graphics C version", Pearson education, Second Edition, 2014.
2. Zhigang Xiang, Roy Plastock, "Computer Graphics", Schaum's outlines, Second Edition, Tata McGraw Hill Edition, 2000.

**REFERENCE BOOKS:**

1. Foley, VanDam, Feiner and Hughes, "Computer Graphics Principles &Practice in C", Pearson Education. 2nd Edition, 1996.
2. David F Rogers, "Procedural elements for Computer Graphics", Tata McGraw Hill, 2nd Edition., 1988.
3. Neuman and Sproul, "Principles of Interactive Computer Graphics, Tata McGraw Hill", 2nd Edition., 1978.
4. Shalini, Govil-Pai, "Principles of Computer Graphics", Springer. 1st Edition, 2006.
5. Steven Harrington, "Computer Graphics", TMH, 2nd Edition, 1987.

# 22CS808 MOBILE APPLICATION DEVELOPMENT

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 2 | 0 | 4 | 4 |

## MOBILE APP DEVELOPMENT



Source: <https://www.tatvasoft.com.au/blog/mobile-application-development-methodology/>

**PREREQUISITE KNOWLEDGE:** OOPs through Java, DDL & DML Commands – DBMS..

### COURSE DESCRIPTION AND OBJECTIVES:

This course guides the student in designing and building a mobile application using Android™. The main objective of this course is to let the student learn basic Android programming concepts while building a variety of apps, starting with basic to making use of advanced concepts.

## MODULE-1

### UNIT-1

**8L+0T+16P=24 Hours**

#### INTRODUCTION:

Introduction to Mobile Application Development - Mobile Applications and Device Platforms, Alternatives for Building Mobile Apps; Introduction to Android, Android versions, Android Architecture.

Application Development Process - Developers Workflow basics, Installing the Android SDK Tools; Anatomy of an Android Application.

Basic Building blocks - Activities, Services, Broadcast Receivers & Content providers; Intents & Fragments.

**View Group- Layout:** Linear Layout, Relative Layout, Frame Layout, Grid Layout, constraint Layout, Table Layout, and Absolute Layout.

### UNIT-2

**8L+0T+16P=24 Hours**

#### VIEWS:

**Views:** Basic Views; Picker Views- Time Picker View, Data Picker View; List Views – List View, Spinner View; Scroll View.

**Activities:** Creating an activity, Understanding the activity life cycle using Log and Toast, applying styles and themes to an activity, and hiding the activity title.

**Linking Activities using Intents:** Introduction to Intents and its types with examples, passing data between activities with intents, Activity Navigation- Implement up navigation with parent activities.

**Fragments:** Introduction to Fragment, the life cycle of a fragment, Adding fragments dynamically, Interaction between fragments.

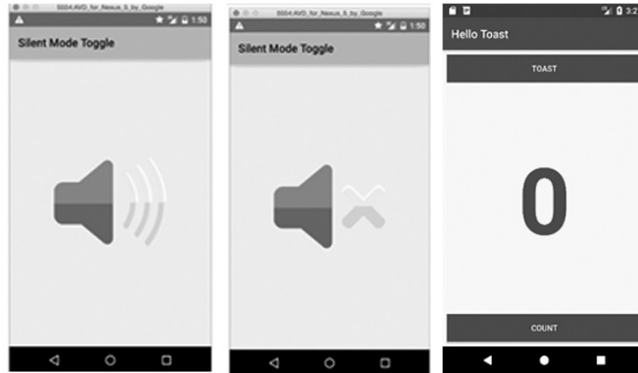
#### PRACTICES:

- Setting up Android Studio:
  - a) Installing Android Studio
  - b) Select an empty activity to simulate the “Welcome App” Using Android Studio.
  - c) Exploring the interface of the Android Studio to understand the Project Structure.
- Develop an Android application using controls like Button, TextView, and EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.
- **Design the HelloToast app:** The HelloToast app consists of two Button elements and one Text View. When the user taps the first Button, it displays a short message (a Toast) on the screen. Tapping the second Button increases a “click” counter displayed in the TextView, which starts at zero.

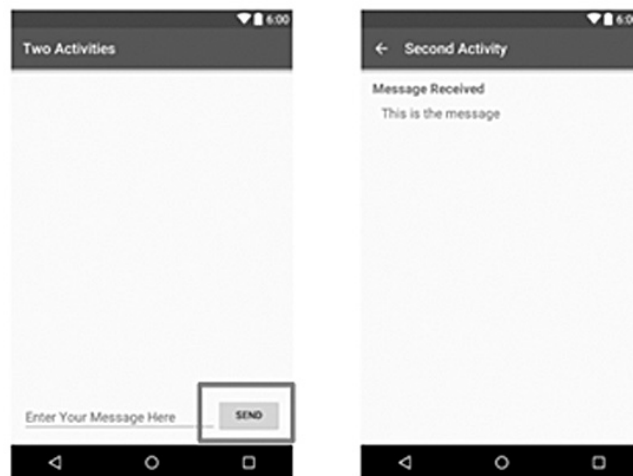
**SKILLS:**

- ✓ *Design mobile applications for user requirements.*
- ✓ *Use of suitable advanced components to design mobile apps.*
- ✓ *Utilization of activities, intents, layouts, and views for content.*

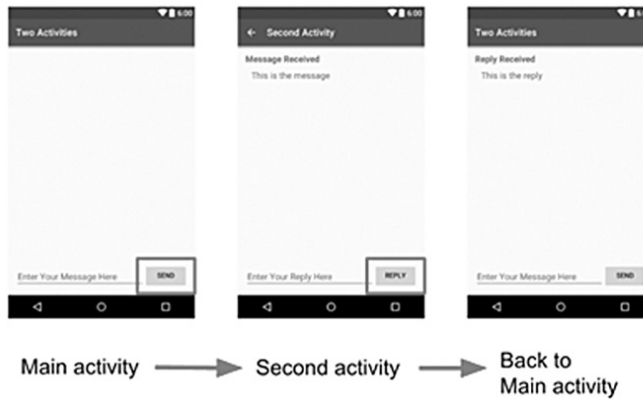
- **Design Silent Mode Toggle application:** This app allows the user to toggle the ringer mode on the phone by simply pressing a button.



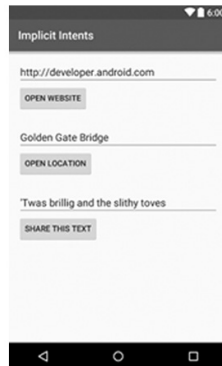
- In this assignment, students will create and build an app called Two Activities. Students will build the app in four stages.
  - a) In the first stage, you create an app whose main activity contains one button, Send. When the user clicks this button, your main activity uses an intent to start the second activity.
  - b) In the second stage, you add an EditText view to the main activity. The user enters a message and clicks Send. The main activity uses an intent to start the second activity and sends the user's message to the second activity. The second activity displays the message received.



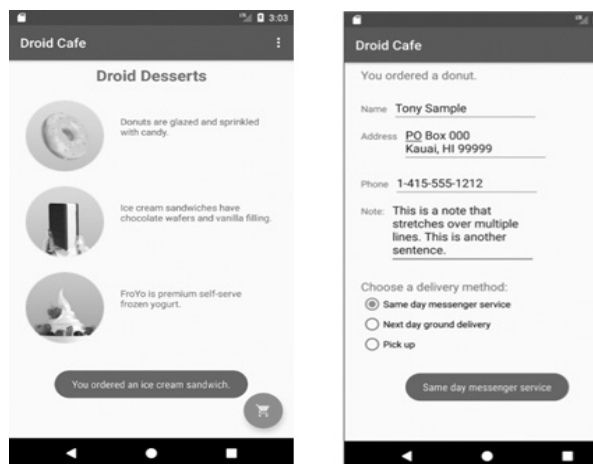
- c) In the final stage of creating the Two Activities app, you add an EditText and a Reply button to the second activity. The user can now type a reply message and tap Reply, and the reply is displayed on the main activity. At this point, you use an intent to pass the reply from the second activity to the main activity.



- **Design an application with implicit intents:** Create a new app with one Activity and three options for actions: open a website, open a location on a map, and share a snippet of text. All the text fields are editable (EditText) but contain default values.



- **Design Droid Café:** In this practical, the student will create and build a new app starting with the Basic Activity template that imitates a dessert-ordering app. The user can tap an image to perform an action-in this case, display a Toast message-as shown in the figure below. The user can also tap a shopping cart button to proceed to the next Activity.
- **Experiment with the android:** inputType attribute for EditText elements. You add EditText elements for a person's name and address and use attributes to define single-line and multiple-line elements that make suggestions as you enter text. You also add an EditText that shows a numeric keypad for entering a phone number.
- Other types of input controls include interactive elements that provide user choices. You add radio buttons to Droid Cafe for choosing only one delivery option from several options. You also offer a spinner input control for selecting the label (Home, Work, Other, Custom) for the phone number.



## MODULE-2

### UNIT-1

8L+0T+16P=24 Hours

#### CREATING A FEATURE-RICH APPLICATION

**Creating a Feature-Rich Application:** Display Orientation – Anchor Views, resizing and repositioning Views, Managing changes to Screen Orientation; Notifications; Action bar; Dialog box; Adapters- Array Adapters and Base Adapters; RecyclerView.

### UNIT-2

8L+0T+16P=24 Hours

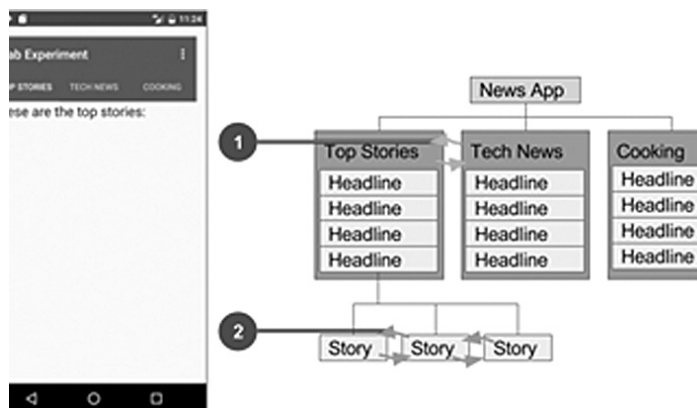
#### SQLITE DATABASE

SQLite Database – Creating the database, Dealing with CRUD;

Firestore-Getting Started with Firestore, Add Firestore to your Android project, Firestore database-Introduction to Firestore database, set up Firestore Real-time Database for Android, Read and Write Data on Android; Publish the App in Play store.

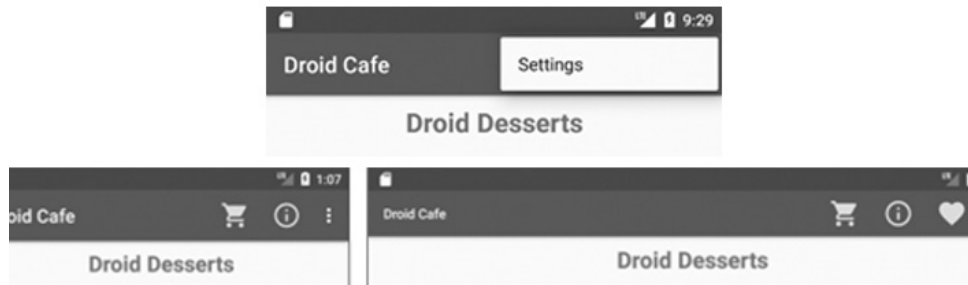
#### PRACTICES:

- Design an application to keep data when the user rotates the device, and when the screen is rotated: When the user rotates the device, Android will normally destroy and re-create the current Activity. You want to keep some data across this cycle, but all the fields in your Activity are lost during it.
- Create a Splash Screen for the existing project- Droid Café from Module-1.
- Design a News App- Consider the following screen as reference:



**NOTE:** Use RecyclerView to display the news under each category.

- Adding more features to Droid Café: In the previous assignments, you created an app called Droid Café, using the Basic Activity template. This template also provides a skeletal options menu in the app bar at the top of the screen.
  - a) Update that menu option as shown in the following images:
  - b) Add notification option: The app must notify the user when the user places the order.
- Provide user authentication for the Droid Café using Firebase – Authentication or SQLite.



- Save all the user preferences in the Firebase Real time Database to fetch whenever required.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                                                                  | Blooms Level | Module No. | Mapping with POs |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply views, intents, and fragments to an existing application.                                                                                  | Apply        | 1          | 2                |
| 2      | Evaluate an existing app to enrich it with new features.                                                                                         | Evaluate     | 2          | 2, 3             |
| 3      | Analyse methods for storing, sharing, and retrieving data in an Android app.                                                                     | Analyse      | 2          | 5                |
| 4      | Design and publish a mobile app in the play store with a database forgiven real-time scenarios using modern tools- Android Studio, and Firebase. | Create       | 2          | 5, 10            |

**TEXT BOOKS:**

1. John Horton “Android Programming for Beginners: Build in-depth, full-featured Android apps starting from zero programming experience”, 3rd Edition, 2021.
2. Wei-Meng Lee, “Beginning Android Application Development”, 1st Edition, John Wiley & Sons, 2012.

**REFERENCE BOOKS:**

1. Michael Burton, “Android App Development for Dummies”, 3rd Edition, A Wiley Brand, 2020.
2. Dawn Griffiths & David Griffiths, “Headfirst Android Development A Brain-Friendly Guide” 2nd Edition, O’Reilly, 2015.
3. <https://aws.amazon.com/mobile/mobile-application-development/>
4. <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/>.





**HONOURS**

**B.Tech.**

# CSE - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

- ▶ 22AM951 - Optimization Techniques
- ▶ 22AM952 - Evolutionary Computing
- ▶ 22AM953 - Kernel Methods for Pattern Analysis
- ▶ 22AM954 - Parallel and Distributed Computing
- ▶ 22AM956 - Capstone Project

**COURSE CONTENTS**

I SEM & II SEM



# 22AM951 OPTIMIZATION TECHNIQUES

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 4 |



Source: <https://oworkers.com/8-process-optimization-techniques-how-to-get-started/>

**PREREQUISITE KNOWLEDGE:** Probability & Linear Algebra.

## COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the fundamental concepts of Optimization Techniques and to make the learners aware of the importance of optimizations in real scenarios. The most important objective is to provide the concepts of various classical and modern methods for constrained and unconstrained problems in both single and multivariable. Finally, the basic idea behind the evolutionary algorithms like Ant Colony Optimization, Particle Swarm Optimization could be discussed for further study.

## MODULE-1

### UNIT-1

12L+8T+0P=20 Hours

#### INTRODUCTION TO CLASSICAL METHODS AND LINEAR PROGRAMMING PROBLEMS TERMINOLOGY:

Design Variables – Constraints, Objective Function - Problem Formulation, Linear Programming Problem - Simplex method - Concept of Duality.

**General Transportation Problem:** The transportation table finding in initial basic feasible solution, North-West corner method, Least cost method, Row minima method, Column minima method.

### UNIT-2

12L+8T+0P=20 Hours

#### SINGLE VARIABLE OPTIMIZATION:

Problems Optimality Criterion - Bracketing Method - Region Elimination Methods -Interval Halving Method - Fibonacci Search Method - Golden Section Method. Gradient Based Methods: Newton -Raphson Method - Bisection Method - Application to Root finding.

**Multi Variable Optimization, Algorithms Optimality:** Criteria - Unidirectional Search. Direct Search Methods: Hooke -Jeeves pattern search method. Gradient Based Methods: Cauchy's Steepest Descent Method - Newton's method, Quasi-Newton methods, L-BFGS.

#### PRACTICES:

- A company makes two products (X and Y) using two machines (A and B). Each unit of X that is produced requires 50 minutes processing time on machine A and 30 minutes processing time on machine B. Each unit of Y that is produced requires 24 minutes processing time on machine A and 33 minutes processing time on machine B. At the start of the current week there are 30 units of X and 90 units of Y in stock. Available processing time on machine A is forecast to be 40 hours and on machine B is forecast to be 35 hours. The demand for X in the current week is forecast to be 75 units and for Y is forecast to be 95 units. Company policy is to maximise the combined sum of the units of X and the units of Y in stock at the end of the week. Formulate the problem of deciding how much of each product to make in the current week as a linear program. Solve this linear program graphically.
- Solve using the Simplex method the following problem:

$$\begin{aligned} \text{Maximize} \quad & Z = f(x,y) = 3x + 2y \\ \text{subject to:} \quad & 2x + y \leq 18 \\ & 2x + 3y \leq 42 \\ & 3x + y \leq 24 \\ & x \geq 0, y \geq 0 \end{aligned}$$

**SKILLS:**

- ✓ *Project scheduling process skills including defining project activities, and estimation of time and resources.*
- ✓ *Resource optimization skills to adjust the project schedule as per the demand and supply issues of project resources.*
- ✓ *Feasibility Analysis.*

- Luminous lamps have three factories - F1, F2, and F3 with production capacity 30, 50, and 20 units per week respectively. These units are to be shipped to four warehouses W1, W2, W3, and W4 with requirement of 20, 40, 30, and 10 units per week respectively. The transportation costs (in Rs.) per unit between factories and warehouses are given below. Find an initial basic feasible solution of the given transportation problem using northwest corner rule

| Factory | Warehouse |    |    |    | Supply |
|---------|-----------|----|----|----|--------|
|         | W1        | W2 | W3 | W4 |        |
| F1      | 1         | 2  | 1  | 4  | 30     |
| F2      | 3         | 3  | 2  | 1  | 50     |
| F3      | 4         | 2  | 5  | 9  | 20     |
| Demand  | 20        | 40 | 30 | 10 |        |

- A mobile phone manufacturing company has three branches located in three different regions, say Jaipur, Udaipur and Mumbai. The company has to transport mobile phones to three destinations, say Kanpur, Pune and Delhi. The availability from Jaipur, Udaipur and Mumbai is 40, 60 and 70 units respectively. The demand at Kanpur, Pune and Delhi are 70, 40 and 60 respectively. The transportation cost is shown in the matrix below (in Rs). Use the Least Cost method to find a basic feasible solution (BFS).

|         |         | Destinations |      |       | Supply |
|---------|---------|--------------|------|-------|--------|
|         |         | Kanpur       | Pune | Delhi |        |
| sources | Jaipur  | 4            | 5    | 1     | 40     |
|         | Udaipur | 3            | 4    | 3     | 60     |
|         | Mumbai  | 6            | 2    | 8     | 70     |
|         | Demand  | 70           | 40   | 60    | 170    |

- Find Solution using Row minima method.

|        | D1  | D2  | D3  | D4  | Supply |
|--------|-----|-----|-----|-----|--------|
| S1     | 11  | 13  | 17  | 14  | 250    |
| S2     | 16  | 18  | 14  | 10  | 300    |
| S3     | 21  | 24  | 13  | 10  | 400    |
| Demand | 200 | 225 | 275 | 250 |        |

- Find Solution using Column minima method.

|        | D1  | D2  | D3  | D4  | Supply |
|--------|-----|-----|-----|-----|--------|
| S1     | 11  | 13  | 17  | 14  | 250    |
| S2     | 16  | 18  | 14  | 10  | 300    |
| S3     | 21  | 24  | 13  | 10  | 400    |
| Demand | 200 | 225 | 275 | 250 |        |

- Find the positive root of the following equation by (Bisection) method.

$$f(x) = \sin\left(\frac{x}{2\pi}\right) - \cos^2 x$$

- Consider figure below. The cross-sectional area A of a gutter with equal base and edge length of 2 is given by:

$$A = 4 \sin \theta (1 + \cos \theta)$$

Using an initial interval of  $[0, \pi / 2]$ , find the interval after 3 iterations. Use an initial interval  $\epsilon = 0.2$ .



## MODULE-2

### UNIT-1

12L+8T+0P=20 Hours

#### STOCHASTIC OPTIMIZATION PROBLEMS:

Notion of regret, online to batch conversion, Methods offering vanishing regret - OGD, EG, OMD

**Convex Sets and Functions:** Affine and convex sets, convexity preserving operations, separating and supporting hyper-planes, generalized inequalities, Operations preserving convexity, conjugate function, Quasi-convex functions, Log-concave and log-convex functions, Convexity with respect to generalized inequalities.

### UNIT-2

12L+8T+0P=20 Hours

#### NON-CONVEX OPTIMIZATION PROBLEMS:

Applications - sparse recovery, affine rank minimization, low-rank matrix completion, Convex approaches - relaxation-based methods, Non-convex approaches - projected gradient descent, alternating minimization

**Algorithms:** Unconstrained problems, equality constrained problems.

#### PRACTICES:

- Show that each convex cone is indeed a convex set.
- Consider the linear system  $0 \leq x_i \leq 1$  for  $i = 1, \dots, n$  and let  $P$  denote the solution set. Explain how to solve a linear programming problem.  $\max\{c^T x : x \in P\}$ .  
What if the linear system was  $a_i \leq x_i \leq b_i$  for  $i = 1, \dots, n$ . Here we assume  $a_i \leq b_i$  for each  $i$ .
- Can projected gradient descent (PGD) be used to obtain a stationary solution?
- Show that strong smoothness does not imply convexity by constructing a nonconvex function  $f: \mathbb{R}^p \rightarrow \mathbb{R}$  that is 1-SS

#### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                | Blooms Level | Module No. | Mapping with POs |
|--------|------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Understand the classical, linear programming and transportation problem along with game theory | Understand   | 1          | 1                |
| 2      | Analyze real world problems around you                                                         | Analyze      | 1          | 1, 2             |
| 3      | Apply optimization to real-world problems                                                      | Apply        | 1          | 1, 2, 3          |
| 4      | Implement dynamic programming in real world complex problems                                   | Implement    | 2          | 3, 4, 11, 12     |

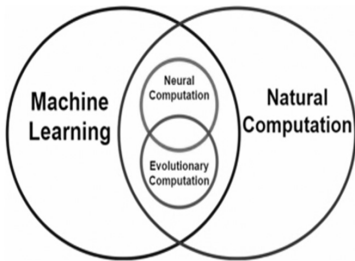
#### TEXT BOOKS:

1. KantiSwarup,ManMohanandP.K.Gupta,“OperationsResearch”,SultanChand&Sons,2005.
2. S. S. Rao, “Engineering Optimization Theory and Practice”, 4th edition, Wiley Publishers,2009.
3. E.J.HaugandJ.S.Arora,“AppliedOptimalDesign”,WileyPublishers,1979.

#### REFERENCES:

1. M.C. Bhuvanewari, “Application of Evolutionary Algorithms for Multi objective Optimization in VLSI and Embedded Systems”, Springer, 2014.
2. Ashlock D, “Evolutionary Computation for Modeling and Optimization”, Springer, 2006.
3. Kalyanmoy Deb, “Optimization for Engineering Design”, Prentice Hall of India, 2nd edition, 2012.
4. A. Ravindran and K.M. Ragsdell, G.V. Reklaites, “Engineering Optimization: Methods and Applications”, Wiley, 2nd edition, 2006.

# 22AM952 EVOLUTIONARY COMPUTING



Source: <https://dl.acm.org/doi/fullHtml/10.1145/3467477>

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 4 |

**PREREQUISITE KNOWLEDGE:** Design and Analysis of Algorithms.

## COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on imparting knowledge about the aspects of Image Processing and its applications. The main objective of the course is to learn digital image fundamentals, image transforms, image enhancement, restoration and compression, morphological image processing, representation and description.

## MODULE-1

### UNIT-1

**12L+8T+0P=20 Hours**

#### INTRODUCTION:

Overview of Philosophy, Nature to Nature Computing, self-Organization, Complexity, Emergence, Bottom-up Vs Top-down approach, determination, chaos and fractals.

**Genetic Algorithms:** Introduction, Representation of Individuals, Mutation, Recombination, Population Models, Survivor Selection.

**Evolution Strategies:** Representation, Self-Adaptation, the Ackley Function, Subjective Evolution of Color Mixes.

### UNIT-2

**12L+8T+0P=20 Hours**

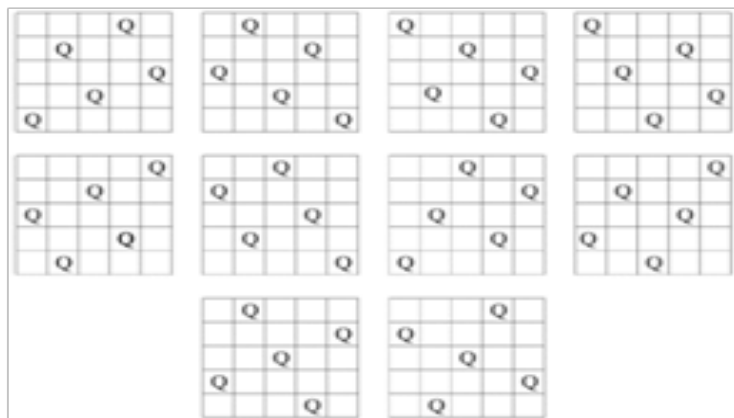
#### GENETIC PROGRAMMING:

Initialization, Bloat in Genetic Programming, Problems Involving "Physical" Environments, Example Application: Symbolic Regression.

Learning Classifier Systems: Criteria - Unidirectional Search. Direct Search Methods: Hooke -Jeeves pattern search method. Gradient Based Methods: Cauchy's Steepest Descent Method - Newton's method, Quasi-Newton methods, L-BFGS.

#### PRACTICES:

- Explain the usage of fitness function in genetic algorithm with example.
- Solve the five queen's problems using Genetic algorithm



Let  $A \in R^{m \times n}$  and  $B \in R^{p \times n}$ . Consider the constrained least squares problem:

minimize  $\|b - Ax\|$

subject to  $Bx = d$

Show that the problem may not have any solution if  $\text{rank}(B) < p$ .

- Use the state-wise regression model to prove that the distance between two valid codes equals the minimum number of small variations to obtain a valid code from another valid code.

## MODULE-2

### UNIT-1

12L+8T+0P=20 Hours

#### PARAMETER CONTROL IN EVOLUTIONARY ALGORITHMS

**Introduction:** Examples of Changing Parameters, Classification of Control Techniques, Examples of Varying EA Parameters.

### UNIT-2

12L+8T+0P=20 Hours

#### MULTIMODAL PROBLEMS AND SPATIAL DISTRIBUTION

**Introduction:** Multimodal Problems and the Need for Diversity, Implicit Measures, Explicit Diversity Maintenance, Multi objective Evolutionary Algorithms, Example Application: Distributed Co-evolution of Job Shop, Introduction to Memetic algorithms.

#### PRACTICES:

- How the global and local aspect of memetic algorithm strategy is used to solve TSP problem?
- How does MOEA help in solving multi-objective optimization problem?

#### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                             | Blooms Level | Module No. | Mapping with POs |
|--------|---------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Understand various evolutionary computation techniques.                                     | Understand   | 1          | 1, 2             |
| 2      | Analyze real world problems and identify evolutionary algorithms suitable for solving them. | Analyze      | 1          | 1, 2, 3          |
| 3      | Apply techniques for optimization, learning, and design.                                    | Apply        | 1          | 1, 2, 3          |
| 4      | Implement evolutionary algorithms in real world complex problems.                           | Implement    | 2          | 11, 12           |

#### TEXT BOOK:

1. A.E. Eiben, J.E Smith, "Introduction to Evolutionary Computing", 2nd edition, Springer, 2015.

#### REFERENCES:

1. Deb, K, "Optimization for Engineering Design", PHI India, 2000.
2. Deb, K. "Multi-objective Optimization using Evolutionary Algorithms", Wiley, UK, 2001.
3. Melanie Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1996.
4. John Koza, "Genetic Programming", MIT Press, 1992.
5. David Fogel, "Evolutionary Computation", The Fossil Record, IEEE Press, 1998.
6. Goldberg D. E, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Massachusetts, 1989.

#### SKILLS:

- ✓ Skills to generate Frameworks of the ideas like Computational, Biology and Computational Psychology
- ✓ Construct biological principles and modeled on natural processes evaluate and apply.

# 22AM953 KERNEL METHODS FOR PATTERN ANALYSIS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**PREREQUISITE KNOWLEDGE:** Machine Learning, Python programming.

## COURSE DESCRIPTION AND OBJECTIVES:

The course kernel methods for pattern analysis (KMPA) deals with the advanced topics in pattern analysis. Pattern analysis mainly involves pattern classification, regression, and pattern clustering. This course primarily focuses on pattern analysis of data that is often not vectors of numbers. For example, protein sequences and structures in computational biology, text and XML documents in web mining, time series in finance, have structures which contain relevant information for the statistical problem but can hardly be encoded into finite-dimensional vector representations. Kernel methods are a class of algorithms well suited for such problems. Indeed, they extend the applicability of many statistical methods initially designed for vectors to virtually any type of data, without the need for explicit vectorization of the data. The goal of this course is to present the mathematical foundations of kernel methods, as well as the main approaches that have emerged so far in kernel design.

## MODULE-1

### UNIT-1

12L+0T+8P=20 Hours

#### INTRODUCTION AND BACKGROUND:

**Introduction to Pattern Analysis:** Pattern analysis tasks, Pattern analysis techniques, Learning methods

**Non-Kernel Methods for Pattern Analysis:** Polynomial curve fitting, Linear model for regression, Regularization, Bias-variance decomposition, Multilayer feedforward neural network, Auto-associative neural network, Radial basis function neural networks.

### UNIT-2

12L+0T+8P=20 Hours

#### DEEP GENERATIVE MODELS:

Boltzmann Machines, Restricted Boltzmann machines, Deep Belief Networks, Deep Boltzmann Machines, Generative Adversarial Networks.

#### PRACTICES:

Implement Models for the following cases:

- Logistic regression model with polynomial basis functions for datasets with (a) Linearly separable classes, (b) Nonlinearly separable classes, (c) Overlapping classes
- MLFFNN with 2 hidden layers using the cross-entropy error function for the datasets with (a) Linearly separable classes, (b) Nonlinearly separable classes, (c) Overlapping classes
- MLFFNN with 3-level stacked RBM based pre-training for Black and white image data
- Visualize Decision region plots for different models specified above
- Plot Confusion matrix for the test data of each dataset and for different models with the best performance.
- Display features extracted by different levels of stacked RBM at the end of pre-training.

## MODULE-2

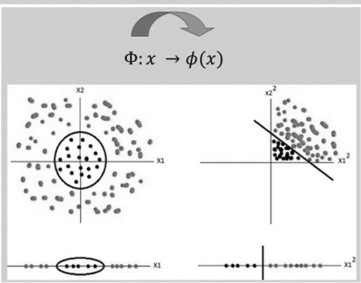
### UNIT-1

14L+0T+10P=24 Hours

#### KERNEL METHODS FOR PATTERN ANALYSIS:

Support vector machine, Support vector regression, Support vector data description, v-support vector methods, Kernel K-means clustering, Kernel principal component analysis, Kernel Fisher discriminant analysis, Kernel canonical correlational analysis.

Source: [https://2.bp.blogspot.com/-Qkk6YNURAEA/W5DrwnnGr5I/AAAAAAAAAD-wQ/SFE6Uj\\_nTwk93ZX-s5wvuhnZQDXFQD833QCLc-BGAs/s1600/m.JPG](https://2.bp.blogspot.com/-Qkk6YNURAEA/W5DrwnnGr5I/AAAAAAAAAD-wQ/SFE6Uj_nTwk93ZX-s5wvuhnZQDXFQD833QCLc-BGAs/s1600/m.JPG)





**Theory of kernels:** Reproducing kernel Hilbert space, The Representer theorem, Mercer's theorem, Operations on kernels, Kernels for structured data: Strings, Sets, Graphs and Trees, Kernel learning methods.

**Theory of kernels:** Reproducing kernel Hilbert space, The Representer theorem, Mercer's theorem, Operations on kernels, Kernels for structured data: Strings, Sets, Graphs and Trees, Kernel learning methods.

## UNIT-2

**10L+0T+16P=16 Hours**

### MACHINE LEARNING PARADIGMS FOR PATTERN ANALYSIS:

**Paradigms in machine learning:** Supervised learning, Semi-Supervised learning, Self-Supervised, Unsupervised learning, and Reinforcement learning. Semi-Supervised learning-Self-training method, Graph-based methods, Semi-supervised Gaussian mixture model, Semi-supervised support vector machine.

### PRACTICES:

- Implement Models for the following cases:
  - Linear kernel based SVM for a dataset with Linearly separable classes
  - Polynomial kernel-based C-SVM for Datasets with (a) Linearly separable classes, (b) Nonlinearly separable classes, (c) Overlapping classes
  - Gaussian kernel-based C-SVM for Datasets with (a) Linearly separable classes, (b) Nonlinearly separable classes, (c) Overlapping classes
  - Visualize Decision region plots for different models specified above
  - Plot Confusion matrix for the test data of each dataset and for different models with the best performance.
  - Images of kernel gram matrices to justify the choice of kernel parameters for the SVMs with best performance, for each kernel and for each dataset as specified above
- Implement Regression Model with v-SVR using Gaussian kernel.
- Implement outlier detection Model with v-SVDD using Gaussian kernel.
- Implement K-means clustering for a 2-dimensional data of nonlinearly separable classes.
- Implement Kernel K-means clustering using Gaussian kernel for a 2-dimensional data of nonlinearly separable classes.
- Implement a Classification using kernels for structured data (Graph representation of data).
- Implement a semi-supervised learning approach for classification.

### COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                       | Blooms Level | Module No. | Mapping with POs |
|--------|---------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Analyze the suitability of kernels for the given supervised and unsupervised problems | Analyze      | 1, 2       | 2, 4             |
| 2      | Designing kernels appropriate for given data                                          | Design       | 1          | 3                |
| 3      | Usage of tools to implement complex models                                            | Apply        | 1, 2       | 1, 5             |
| 4      | Application of kernels for pattern analysis                                           | Apply        | 1, 2       | 1                |

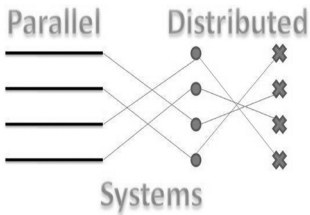
### TEXT BOOKS:

1. B. Scholkopf and A.J.Smola, "Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond", The MIT Press, 2002
2. Ian Good fellow and Yoshua Bengio and Aaron, "Deep Learning", 1st Edition, An MIT Press Book, 2016.

### REFERENCE LINKS:

1. S. Haykin, "Neural Networks and Learning Machines", 3rd edition, Prentice Hall of India, 2011.
2. Satish Kumar, "Neural Networks, A Classroom Approach", Tata McGraw -Hill, 2007.
3. C.M.Bishop, "Pattern Recognition and Machine Learning", Cambridge University Press, 2006.
4. V.Vapnik, "Statistical Learning Theory", John Wiley & Sons, 1999.

# 22AM954 PARALLEL AND DISTRIBUTED COMPUTING



Source: <http://uceou.edu/PDS/About%20PDS.html>

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | 2 | 0 | 4 |

**PREREQUISITE KNOWLEDGE:** Fundamental of programming and operating system, computer networking, and computer Organization.

## COURSE DESCRIPTION AND OBJECTIVES:

This course covers the challenges faced in constructing parallel and distributed applications, Various implementation techniques, paradigms, architectures and parallel algorithms. And current trends in parallel and distributed computing like Open MP, POSIX Threads, Apache Hadoop (DFS).

## MODULE-1

### UNIT-1

**12L+8T+0P=20 Hours**

#### INTRODUCTION TO PARALLEL COMPUTING:

The Idea of Parallelism, Power and potential of parallelism, examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instructions streams.

**Parallel Architecture:** Pipeline architecture, Array processor, Multi-processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, Memory Issues: Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, Vector processing, GPU co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

### UNIT-2

**12L+8T+0P=20 Hours**

#### PARALLEL ALGORITHM DESIGN PRINCIPLES AND PROGRAMMING:

Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task- Based Decomposition, Data Parallel Decomposition, Characteristics of task and interaction, Load balancing, Data Management, parallel algorithm models, Sources of overhead in parallel programs, Performance metrics for parallel algorithm implementations, Parallel algorithmic patterns like divide and conquer, Map and Reduce, Specific algorithms like parallel Merge Sort, Parallel graph Algorithms.

#### PRACTICES:

- Identify Multiple Instruction Single Data, or MISD. How would an MISD system work? Give an example.
- Suppose a shared-memory system uses snooping cache coherence and write-back caches. Also suppose that core 0 has the variable x in its cache, and it executes the assignment  $x = 5$ . Finally suppose that core 1 doesn't have x in its cache, and after core 0's update to x, core 1 tries to execute  $y = x$ . What value will be assigned to y? Why?
- Consider a simplified version of bucket-sort. You are given an array A of n random integers in the range  $[1...r]$  as input. The output data consist of r buckets, such that at the end of the algorithm, Bucket i contains indices of all the elements in A that are equal to i.
  - Describe a decomposition based on partitioning the input data (i.e., the array A) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.

- Describe a decomposition based on partitioning the output data (i.e., the set of  $r$  buckets) and an appropriate mapping onto  $p$  processes. Describe briefly how the resulting parallel algorithm would work.
- Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming that it does not take any time to assign work to a process, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

## MODULE-2

### UNIT-1

12L+8T+0P=20 Hours

#### INTRODUCTION TO DISTRIBUTED SYSTEMS:

Goals of the Distributed Systems, Relation to parallel systems, synchronous versus asynchronous execution, design issues and challenges, Types of Distributed Systems, Distributed System Models, Hardware and software concepts related to distributed systems, middleware models.

**Distributed Computing and Communication design principles:** A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- 08 Passing, Consensus, Conditional Actions, Critical Paths, Scalability, and cache coherence in multiprocessor systems, synchronization mechanism.

### UNIT-2

12L+8T+0P=20 Hours

#### PARALLEL AND DISTRIBUTED PROGRAMMING FRAMEWORKS:

Overview of CUDA, OpenMP, POSIX Threads, Apache Hadoop (DFS), and current trends in parallel and distributed computing.

#### PRACTICES:

- Give five types of hardware resource and five types of data or software resource that can usefully be shared. Give examples of their sharing as it occurs in practice in distributed systems.
- The host computers used in peer-to-peer systems are often simply desktop computers in users' offices or homes. What are the implications of this for the availability and security of any shared data objects that they hold and to what extent can any weaknesses be overcome through the use of replication?
- Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed or delivered too fast for the recipient to handle them, but those that are delivered arrive with the correct contents. Describe the classes of failure exhibited by each service. Classify their failures according to their effects on the properties of validity and integrity. Can service B be described as a reliable communication service?
- Illustrate distributed design through a substantial case study, examining in detail the design of the Google infrastructure, a platform and associated middleware that supports both Google search and a set of associated web services and applications including Google Apps.
- Implementation of the parallel algorithms (on a PC-cluster under Linux platform). The programs will be based on POSIX Thread, MPI programming, Hadoop, Apache Spark etc.

#### SKILLS:

- ✓ Recognize parallelism in computational problems.
- ✓ Know different parallel systems and their classification.
- ✓ Design parallel algorithms for different applications.
- ✓ Compare replication schemes with respect to performance, availability and consistency concerns.
- ✓ Design, implement, and debug distributed systems.
- ✓ Implement parallel algorithms using MPI and Open MP environments.

**COURSE OUTCOMES:**

Upon successful completion of this course, students will have the ability to:

| CO No. | Course Outcomes                                                                                           | Blooms Level | Module No. | Mapping with POs |
|--------|-----------------------------------------------------------------------------------------------------------|--------------|------------|------------------|
| 1      | Apply the fundamentals of parallel and parallel computing including architectures and paradigms.          | Apply        | 1          | 1                |
| 2      | Analyse the various design principles of parallel algorithms.                                             | Analyse      | 1          | 2                |
| 3      | Learn the intricacies of distributed programming                                                          | Understand   | 2          | 1                |
| 4      | Develop and execute basic parallel and distributed applications using basic programming models and tools. | create       | 2          | 5                |

**TEXT BOOKS:**

1. Arun Kulkarni, Nupur PrasadGiri, Nikhilesh Joshi, Bhushan Jadhav, "Parallel and Distributed Systems" 2nd Edition, Wiley, 2017.
2. Ananth Grama, Anshul Gupta, and George Karypis, Vipin Kumar, "Introduction to Parallel Computing", 2nd Edition,Addition Wesley, 2003.
3. Distributed Systems Principles and Paradigms Andrew S. Tanenbaum Maarten Van Steen, 3rd Edition, 2017.

**REFERENCE BOOKS:**

1. DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition. George Coulouris. Cambridge University. Jean Dollimore formerly of Queen Mary,University of London.
2. Introduction To Parallel Programming, Peter S. Pacheco University of San Francisco.
3. Introduction To Parallel Processing, M.Sasikumar, Dinesh Shikhare and P. Ravi Prakash, Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley.
4. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
5. Parallel Programming in C with MPI and OpenMP Michael J.Quinn, McGrawHill Higher Education.

**REFERENCE LINKS:**

1. <https://hpc.llnl.gov/training/tutorials/introduction-parallel-computing-tutorial>
2. <https://www.geeksforgeeks.org/introduction-to-parallel-computing/>
3. <https://nptel.ac.in/>
4. <https://www.coursera.org/>