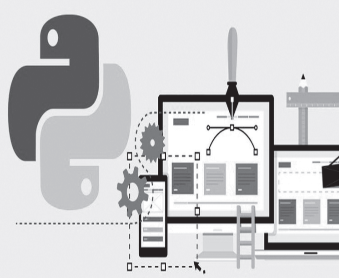
DEPARTMENT ELECTIVES

B.Tech.

COURSE CONTENTS

19CS331 PYTHON PROGRAMMING



source:
https://
i.udemycdn.com/

Hours Per Week :

L	T	P	C
3	0	2	4

Total Hours :

L	T	P	CS	WA/RA	SSH	SA	S	BS
45	-	30	5	5	30	20	5	5

COURSE DESCRIPTION AND OBJECTIVES :

This course offers sufficient knowledge required to understand the fundamental concepts of Python programming language. This course enable the students to use different data structures like lists, dictionaries, tuples, sets etc. This course also enable the students to create reliable, modular and reusable programming and to create applications using Object-Oriented Programming approach.

COURSE OUTCOMES :

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Analyze the usage of different data structures for practical and contemporary applications for a given problem.	1
2	Develop functional, reliable and user friendly Python programs for given problem statement and constraints.	2
3	Installing the python environment and related packages that are required for practical and contemporary applications.	3
4	Design programs using the concepts of object oriented programming paradigm.	4
5	Create simple programming solutions to the given problems.	5

SKILLS:

- ✓ *Identify suitable data types and data structures required for an application.*
- ✓ *Design structured and Object oriented programming solutions.*
- ✓ *Design reliable and modular applications for a given problem.*

UNIT– I **L- 9**

INTRODUCTION: History of python, Features of python, Python installation on windows & Linux, Installing python packages via PIP, Running python commands using the REPL(Shell), Running python scripts, Variables, Assignment, Keywords, Input-output, Indentation, Basic data types - integers, booleans etc.

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

UNIT – II **L- 9**

CONTROL STRUCTURES: Conditional control structures - if, elif, else; Loop control structures - for, while, for... else, while..else, nested loops, break, continue, pass.

PYTHON DATA STRUCTURES: Lists, Tuples, Dictionary - creation, accessing, basic operators and methods.

UNIT – III **L- 9**

OTHER DATA STRUCTURES: Strings - creation, accessing, operators, methods; Sets - creation, accessing, operators, methods; List comprehensions; Functions - defining functions, calling functions, passing arguments - keyword arguments, default arguments, variable-length arguments, anonymous functions (lambda), fruitful functions (function returning values), scope of the variables in a function - global and local variables.

UNIT – IV **L- 9**

MODULES: Creating modules, Import statement, From...import statement, Name spacing.

ERRORS AND EXCEPTIONS: Difference between an error and exception, Handling exception, Try except block, Raising exceptions, User defined exceptions.

FILE PROCESSING: Reading and Writing Files - creating a new file, writing to a file, reading files as text, opening and closing files, reading and writing, tell(), seek(), rename().

UNIT - V **L- 9**

OBJECT ORIENTED PROGRAMMING IN PYTHON: Classes, ‘self variable’ methods, Constructor method, Inheritance, Overriding methods, Data hiding.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Given an integer, n, depending on the value of n display suitable message:

   If n is odd, print Great.

   If n is even and in the inclusive range of 10 to 20 , print Not Great.

   If n is even and in the inclusive range of 21 to 30, print Great.

   If n is even and greater than 30, print Not Great.

2. Read integers a, b from standard input and display the following in different lines:

   The first line contains product of a and b.

   The second line contains the difference of the two numbers (a - b).

   The third line contains the exp(a,b).

   The fourth line contains the integer division of a, b.

   The fifth line contains the float division of a, b.

3. Read an integer n and display the following:

   1,2,3,4…….n.

   123…….n (without any spaces).

   1 2 3…….n (separated by spaces).

   1, 4, 9, 16, …….n^2.

4. Use list comprehensions to solve Exercise 3.

5. Given the mid1 marks of python course for all the students, you are required to find the second highest mark. (Use lists).

6. Create two lists of integer numbers and write python code to generate a new list as described below (Make sure your code works on two lists of different sizes)

   a. A list with elements that are common in both the lists. (print without duplicates).

   b. A list with elements that are in either of the lists.

   c. A list that contains the first eleent of the first list and last element of the second list.

   d. A list that contains sum of elements of first list and sum of elements of second list.

   e. A list that contains largest number of both the lists.

   f. A list that contains least number of both the lists.

7. Use a dictionary to store the details of student name and marks in 3 subjects (data structures, DAA and python). Now display the details of students in ascending order of marks scored in python.

8. Write a Python program to generate a list, whose elements are tuples. Then display the list with tuples sorted in increasing order by the last element of each tuple.

   Sample Input: [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]
   Expected output : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

9. Write a program (function!) that reads a list and create a new list that contains all the elements of the original list without any duplicates.
   Write two different functions to do this - one using a loop and constructing a list, and another using sets.

10. Given a string and a substring, print the number of times that the substring occurs in the given string.

11. Read a string and write different python functions to achieve the following:

    a) To remove vowels in the given string using control transfer statements.

    b) To count number of uppercase and lowercase letters in the given string.

    c) To remove all punctuation characters from given string.

    d) To check whether the given string is palindrome or not.

    e) To swap case of each letter in the string.

12. Write a python function to read a text document and do the following:

    a) Count number of words in a given text.

    b) To display the words in alphabetic order.

    c) To count number of sentences in the text.

    d) To display words of the document in reverse order.

    e) Count the number of words with 3 characters.

13. Write a Python program to read data from two different text files. Display a line from first file followed by the corresponding line in the second file.

14. a. Define a function simple_int(p, t, r) that accepts 3 arguments and returns simple interest accordingly. call the function simple_int(p, t, r) with positional parameters.

    b. Define a function simple_int(p, t, r) that accepts 3 arguments and returns simple interest accordingly. Call the function simple_int(p,t,r) with keyword arguments where the order of arguments does not matter.

    c. Define a function simple_int(p, t, r=val) where r is set with a default value. Call the function simple_int() and pass all 3 parameters, pass only 2 parameters, pass the parameters in different order in function call.

    d. Define a function mul(*args) which can handle variable length arguments. Call the function with 2, 5 and 7 parameters.

15. write 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.

16. Define the following functions that are more robust to erroneous input/data

    a) To divide two numbers (To handle ZeroDivisionError).

    b) To read two integer numbers and display them (To handle ValueError).

    c) To display elements of a list (To handle IndexError).

    d) To open a file and display file contents (To handle FileNotFoundError).

17. Write python code to handle multiple exceptions and to raise an exception manually.

18. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle. Define parameterized and non-parameterized constructors.

19. Write Python code to depict the following oops concepts:

    a) Data hiding.

    b) Inheritance.

    c) Overriding.

20. Define a class Person and its two child classes: Male & Female. All classes have method "get Gender" which can print "Male" for Male class and "Female" for female class.

TEXT BOOKS:

1. Vamsi Kurama, "Python Programming: A Modern Approach", 1st edition, Pearson Publishers, 2018.

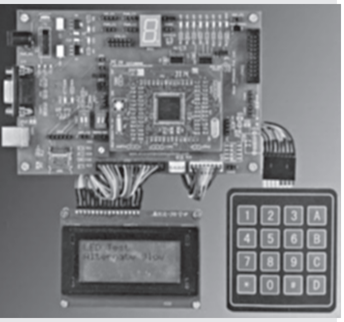
2. Mark Lutz, "Learning Python", 5th edition, Orielly Publishers, 2013.

REFERENCE BOOKS:

1. Allen Downey, "Think Python", 2nd edition, Green Tea Press, 2016.

2. Ashok Namdev Kamthane and Amith Ashok Kamthane, "Programming and Problem Solving with Python", 1st edition, McGraw Hill Education, 2016.

3. W. J. Chun, "Core Python Programming", 3rd edition, Pearson Publishers, 2013.



source:
https://5.imimg.com/

19CS332 EMBEDDED SYSTEMS

Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSES: Operating Systems; Computer Organization.

COURSE DESCRIPTION AND OBJECTIVES:

This course emphasizes on comprehensive treatment of embedded hardware and real time operating systems along with case studies, in tune with the requirements of Industry. The objective of this course is to enable the students to understand embedded-system programming and apply that knowledge to design and develop embedded solutions.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand the concept of embedded system, microcontroller and Real-time operating system.	1
2	Differentiate various components of microcontroller and their interactions.	2
3	Make use of programming environment in ARM to develop embedded solutions.	3,10,12
4	Deployment of embedded software into target system. Graph based problems.	3,5

SKILLS:

- ✓ Programming the ARM processors.
- ✓ Design of microcontroller based embedded system.
- ✓ Interface of various peripherals with ARM processors.
- ✓ Expertise in writing multiple tasks under RTOS environment.
- ✓ Handle shared data issues in RTOS environment.

UNIT– I **L- 8**

INTRODUCTION TO EMBEDDED SYSTEMS: Definition; Applications of ES; Embedded hardware units and devices; Embedded Software; Design process - design metrics in ES, challenges in ES design.

UNIT – II **L- 10**

ARCHITECTURE OF 8051: 8051 Micro controller hardware; Input/output ports and circuits; External memory; Counter and timers; Serial data input/output; Interrupts.

UNIT – III **L- 9**

ARM- EMBEDDED PROCESSOR: History; Architecture; Interrupt vector; Programming the ARM - ARM assembly language; Instruction set, Conditional execution; Arithmetic and logical compare.

UNIT – IV **L- 10**

ARM PROGRAMMING: Assembly programming; General structure of assembly language; Writing programs; Branch instructions; Loading constrains; Load and store instructions; Read-only and read/write memory; Multiple register load and store.

UNIT - V **L- 8**

REAL TIME OPERATING SYSTEMS: RTOS introduction; Tasks and task states; Tasks and data; Reentrancy; Semaphores and shared data; Embedded software development tools - host and target machines, linker/locators for embedded software, getting embedded software into the target system.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Write a program to:
   - a. implement arithmetic and logical operations.
   - b. perform control operations.
   - c. find sum of n natural numbers.
   - d. evaluate the value of the given series.
   - e. count 0's and 1's in a given register.
   - f. perform 24-bit shift.
   - g. arrange values of n registers in ascending and descending order.
2. Identify the status of switches using I/O interface.
3. Write a program for serial communication.
4. Write a program for encryption/ decryption.
5. Write a program to process the data provided by the sensors using 8051 and display thedata on PC monitor.
6. Port RTOS (MuCOS) on to 89C51 board and verify.
7. 4-digit, 7-segment LED display on ESA MCB 51.
8. Stepper motor on ESA MCB 51.
9. Traffic lights on ESA MCB 51.
10. Simulate an elevator movement using RTOS on 89C51 board.
11. Familiarization of ARM programming model using ARM kit.
12. Take parallel input from port P1 convert it into serial and send it via PO.

TEXT BOOKS:

1. Raj Kamal, "Embedded Systems", Tata McGraw Hill, 2nd edition, 2009.
2. Lyla B Das, "Embedded Systems an Integrated Approach", Pearson Education, 1st edition, 2012.
3. David E. Simon, "An Embedded Software Primer", Pearson Education,1st edition, 2008.

REFERENCE BOOKS:

1. Wayne Wolf, "Computers as Components-Principles of Embedded Computer System Design", Elseveir, 1st edition, 2009.
2. Labrosse, "Embedding System Building Blocks", 2nd edition, CMP Publishers, 2007.

19CS333 OPEN SOURCE WEB TECHNOLOGIES

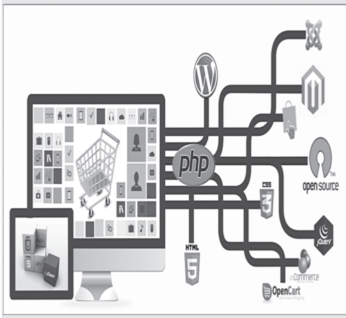
Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5



sourcee:
http://www. comval.in/

PREREQUISITE COURSES: Web Technologies, Database Management System.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the concepts pertaining to open source technologies such as LINUX, MySQL, PHP, Apache web server and various other tools used to develop web applications. The objective of this course is to offer insight into various open source technologies to develop web applications.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Analyze various opensource and commercial products.	2
2	Design dynamic web pages, Web services using PHP.	3
3	Apply HTML5 tags for web page design.	1
4	Develop web based applications.	3

SKILLS:

- ✓ *Develop web applications using open source technologies.*
- ✓ *Develop web applications for Linux.*
- ✓ *Generate dynamic contents using PHP programming language.*
- ✓ *Design and develop database oriented application using PHP and MySQL.*
- ✓ *Apache web server administration.*

UNIT– I **L- 9**

OPEN SOURCE: Introduction, Open source operating System, Nature of open sources, Advantages, Application of open sources.

Open Source Programming Languages: Introduction to dynamic web content, Setting up a development server, Introduction to PHP, Expressions and control flow in PHP, PHP functions and Objects, PHP arrays, Practical PHP.

UNIT – II **L- 9**

OPEN SOURCE DATABASE: Introduction to MySQL, Mastering MySQL, Accessing MySQL using PHP-querying a MySQL database with PHP, Practical MySQL, Preventing hacking attempts, Using MySQL procedure.

UNIT – III **L- 9**

FORM HANDLING: Form Handling building forms, Retrieving submitted data, An example program, Cookies, Sessions and authentication using cookies in PHP, HTTP authentication, using sessions.

UNIT – IV **L- 9**

HTML5: Introduction to HTML5, The HTML5 Canvas, HTML5 audio and video, Bringing it all together-designing a social networking site.

UNIT - V **L- 9**

ANGULAR JS: The basics of AngularJS, Introduction MVC, Filters and modules, Directives, Working with Forms, Services and server communication, Organizing views, Angular JS animation, Deployment considerations.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Study of LINUX operating system and Web Servers in detail.
2. Write a PHP program:
   a. To print Hello World.
   b. To add two numbers.
   c. To Swap two numbers.
   d. To print Even and Odd Numbers.
3. Write a PHP program:
   a. To print factorial of a number.
   b. To print Armstrong number.
   c. To print Palindrome.
   d. To print Prime numbers.
   e. To print Fibonacci numbers.
   f. To print Reverse of a number and String.
   g. To print Table.
4. Write a PHP program to perform PHP string methods.
5. Write a PHP program to perform user defined functions and arrays.
6. Write a PHP program to create user registration form and include validations.
7. Write a PHP program to demonstrate using COOKIES and SESSIONS using forms.
8. Write a PHP program to demonstrate data manipulation using PHP & MySQL.
9. Write a PHP program to demonstrate data manipulation using PHP & MySQL.
10. Develop a Social Networking site with security options.
11. Develop a Search Engine (use MySQL as backend, HTML5 as a frontend and PHP as a middleware).
12. Develop a web application using AngularJS.

TEXT BOOKS:

1. Robin Nixon, “Learning PHP, MySQL & JavaScript WITH JQUERY, CSS & HTML5”, 4th edition, O’Reilly, 2015.
2. Andrew Grant, “Beginning Angular JS”, 1st edition, Apress, 2014.

REFERENCE BOOKS:

1. Steve Prettyman, “Learn PHP 7 Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL”, 1st edition, Apress, 2015.
2. 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.

19CS334 FUNDAMENTALS OF IMAGE PROCESSING

Source: http://www.e2matrix.com/

Hours Per Week :

L	T	P	C
3	0	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PRE-REQUISITE COURSES : Engineering Mathematics; Probability and 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.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Apply various compression techniques to reduce image size and morphological operations to extract features.	1
2	Analyse images in the frequency domain using various transforms.	2
3	Evaluate the techniques for image enhancement and image restoration.	4
4	Interpret Image compression standards, segmentation and representation techniques.	4

SKILLS:

- ✓ *Apply knowledge of science and engineering principles to image related problems.*
- ✓ *Undertake image problem identification and formulate solutions.*

UNIT– I **L- 9**

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 – II **L- 9**

IMAGE ENHANCEMENT IN THE SPATIAL AND FREQUENCY DOMAINS: Basic gray - level transformations, Histogram processing, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters, The basics of filtering in the frequency domain, Image smoothing in frequency domain filters, Image sharpening in frequency domain filters.

UNIT – III **L- 9**

IMAGE SEGMENTATION: Fundamentals, Point, Line and edge detection, Thresholding, Region-based segmentation, Segmentation using morphological watersheds, The use of motion in segmentation.

UNIT – IV **L- 9**

IMAGE COMPRESSION: Fundamentals, Huffman coding, Golomb coding, LZW coding, Run-length coding, Bit-plane coding, Block transform coding, Predictive coding, Wavelet coding, Image compression standards.

UNIT - V **L- 9**

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.

Representation and description: Chain codes, Polygonal approximation, Signature, Boundary segments, Skeletons, Boundary descriptors, Regional descriptors.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Program to enhance image using image arithmetic and logical operations.
2. Program for an image enhancement using pixel operation.
3. Program for gray level slicing with and without background.
4. Program for image enhancement using histogram equalization.
5. Program to filter an image using averaging low pass filter in spatial domain and median filter.
6. Program to sharpen an image using 2-D laplacian high pass filter in spatial domain.
7. Program for detecting edges in an image using Roberts cross gradient operator and sobel operator.
8. Program for smooth an image using low pass filter in frequency domain. (Butterworth lpf).
9. Program for smooth an image using high pass filter in frequency domain. (Butterworth hpf).
10. Program for morphological image operations-erosion, dilation, opening & closing.
11. Program for image compression.
12. Program for image restoration.
13. Program for image segmentation.

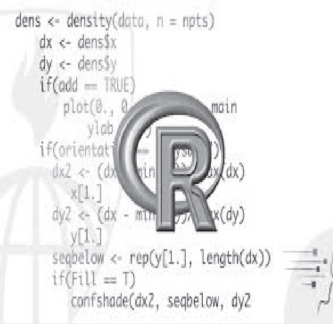
TEXT BOOK:

1. Rafeal C Gonzalez and Richard E.Woods, “Digital Image Processing”, 3rd edition, Pearson Education/ PHI, 2008.

REFERENCE BOOKS:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image Processing, Analysis and Machine Vision”, 4th edition, Cengage, 2015.
2. Alasdair McAndrew, “Introduction to Digital Image Processing with Matlab”, Thomson Course Technology, 2004 Course Technology Press, Boston, MA, United States, 2004.
3. William K. Prat, “Digital Image Processing”, 4th edition, Wiley-Interscience, A John Wiley & Sons, Inc., Publication, 2007.

19CS335 R PROGRAMMING



source: https://d3njjcbhbojbot.cloudfront.net/api/

Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

COURSE DESCRIPTION AND OBJECTIVES:

R is a programming language that is capable of handling mathematical and statistical manipulations. It has its own programming language as well as built in functions to perform any specialized task. After taking the course, students will be able to familiar with the use of the R interactive environment, Understand the different data types, data structures, analyze the data by using different techniques. In addition to this the student will be able to display the data by using different graphics.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Apply different data structures for solving a program.	1
2	Analyze the data by applying both linear and non linear regression techniques.	2
3	Investigate results obtained for a given data set by using different plots.	4
4	Design and develop a program for a given scenario.	3,5,8,9,10

SKILLS:

- ✓ *Perform basic matrix operations such as addition, subtraction, dot product and multiplication on matrices.*
- ✓ *Write programs for a given scenario.*
- ✓ *Predict the things by using classification techniques.*

UNIT– I **L- 9**

INTRODUCTION: Install R & R studio, opening R, The R user interface; Objects; Functions; Sample with replacements; Writing your own functions - The function constructor, arguments, scripts.

R Objects: Atomic vectors; Attributes; Matrices; Arrays.

UNIT – II **L- 9**

R OBJECTS & R NOTATIONS: Class; Coercion; Lists; Data frames; Loading data; Saving data.
R Notation: Selecting values; Dollar signs and double brackets.

UNIT – III **L- 9**

MODIFYING VALUES & PROGRAMS AND LOOPS: Changing values in place; Logical sub setting - logical tests, Boolean operators; Missing information - na.rm, is.na;

Programs: Strategy; Sequential steps; Parallel cases; If statements; Else statements, Lookup tables; Code comments.
Loops: Expected values; Expand. grid; For loops; While loops; Repeat loops.

UNIT – IV **L- 9**

GRAPHICS: Creating graphs; The workhorse of R base graphics; The plot() Function; Customizing Graphs; Saving graphs to files.

R Packages & Loading and Saving Data in R: R packages - installing packages, loading packages; Loading and saving Data in R; Data sets in base R; Working directory.

UNIT - V **L- 9**

REGRESSION: Linear models; Simple linear regression; Multiple regression; Generalized linear models - Logistic regression, poisson regression; Other generalized linear models; Survival analysis; Nonlinear models; Splines; Decision; Random forests.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. (a) Installation of R.

   (b) Implement R program for finding the factorial of a given number.

2. (a)Implement R program to read two integers and perform all possible arithmetic operations and display the results.

   (b)Write R program to create a vector which contains 10 random integer values between -50 and +50.

   (c) Write R program to create three vectors having numeric data, character data and logical data. Display the content of the vectors and their type.

3. Write R program to create data frame which contain details of employees such as name, salary and date of birth. Display statistics of the data on salary and date of birth.

4. Write R program to create and print a list of elements using vectors, matrices.

5. Write R program to create and print list of heterogeneous data, which include character, numeric and logical vectors.

6. Write R program to read CSV file and perform the following preprocessing operations:

   (a)Display the contents of CSV file.

   (b) Find the missing values in a dataset.

   (c) How to remove missing values in a data set.

7. Write R program to find frequent item sets for a data set using Apriori algorithm.

8. Write R program for clustering of given data sets using the following clustering algorithms and compare their performance

   (a)K means (b) K Medoids

9. Write R program for classification of a given datasets using the following classification algorithms and compare the classification accuracy.

   (a) KNN (b) Naive Bayes (c) Decision tree

10. Implement R program to import packages and create a bar plot and bell curve of a random normal distribution for a given data.

11. Write R program to predict the value of dependent variable for a given independent variable data and find out the relationship among the variables using regression.

TEXT BOOKS:

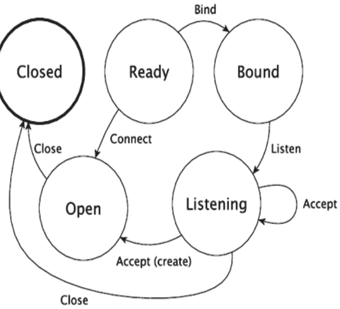
1. Garrett Grolemund, “Hands-on Programming with R”, 1st edition, O’Reilly Press, 2014.

2. Norman Matloff, “The Art of R Programming”, 1st edition, No Starch Press, 2017.

3. Johannes Ledolter, “Data Mining and Business Analytics with R”, 1st edition, Wiley publishers, 2014.

REFERENCE BOOK:

1. Michael J, Crawley, “The R Book”, 1st edition, Wiley Publishers, 2012.



source: http://docs.idris-lang.org/

19CS336 NETWORK PROGRAMMING

Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSE: Computer Networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course will cover the practical aspects of computer network programming, with emphasis on the Internet. The goal of this course is to introduce the students to the basics of computer networks and Internet programming.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Identify the various standards of networking models to express the mathematical properties.	1
2	Develop the underlying Inter Process Communication and Remote Login.	2
3	Identify different protocols of each OSI layer.	3
4	Compare the conventional sockets with UDP sockets process.	4
5	Analyse TCP Client and Server process code concepts with I/O Multiplexing and sockets Operations.	3

SKILLS:

- ✓ *Establish local area networks with different topologies.*
- ✓ *Evaluate various routing protocols.*
- ✓ *Concentrate on Unix Inter Process Communication.*
- ✓ *Network trouble shooting such as installing network interface card drivers, setting IP addresses, and subnet masking etc.*
- ✓ *Apache web server administration.*

UNIT– I **L- 9**

INTRODUCTION TO NETWORK PROGRAMMING: OSI model, Unix standards, TCP and UDP, TCP connection establishment and format, Buffer sizes and limitation, Standard internet services, Protocol usage by common internet application.

UNIT – II **L- 9**

SOCKETS: Address structures, Value - result arguments, byte ordering and manipulation function and related functions; Elementary TCP sockets - socket, connect, bind, listen, accept, fork and exec function, concurrent servers; Close function and related function.

TCP client server: Introduction, TCP Echo server functions, Normal start-up, terminate and signal handling server process termination, Crashing and rebooting of server host, shutdown of server host.

UNIT – III **L- 9**

I/O MULTIPLEXING AND SOCKET OPTIONS: I/O Models, Select function, Batch input, Shutdown function, Poll function, TCP Echo server - getsockopt and setsockopt functions; Socket states, Generic socket option, IPV6 socket option, ICMPV6 socket option, IPV6 socket option and TCP socket options.

UNIT – IV **L- 9**

ELEMENTARY UDP SOCKETS: Introduction UDP Echo server function, Lost datagram, Summary of UDP example, Lack of flow control with UDP, Determining outgoing interface with UDP; Elementary name and address conversions - DNS, gethost by name function, resolver option, function and IPV6 support, uname function, other networking information.

UNIT - V **L- 9**

IPC: Introduction, File and record locking, Pipes, FIFOs streams and messages, Name spaces, System IPC, Message queues, Semaphores; Remote login - terminal line disciplines, pseudo-terminals, terminal modes, control terminals, rlogin overview, RPC transparency Issues.

TEXT BOOKS:

1. W R Stevens, “Unix Network Programming, Inter Process Communication”, 2nd edition, Prentice Hall of India/Pearson, 2014.
2. W R Stevens, “Unix Network Programming, the Sockets Networking API”, 3rd edition, Prentice Hall of India/Pearson, 2012.
3. Andrew S Tanenbaum, “Computer Networks”, 5th edition. Pearson Education, 2014.

REFERENCE BOOKS:

1. N Matthew, R Stones and Wrox, “Beginning Linux Programming”, 4th edition, Wiley India Edition, 2008.
2. Robert Love, “Linux System Programming”, 2nd edition, O’Reilly, 2013.
3. Graham Glass and King Ables, “Unix for programmers and users”, 3rd edition, Pearson Education, 2003.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Implement the following forms of IPC.

   a) Pipes b) FIFO

2. Implement file transfer using Message Queue form of IPC.
3. Write a Program to create an integer variable using Shared Memory concept and increment the variable simultaneously by two processes. Use Semaphores to avoid Race conditions.
4. Design TCP iterative Client - Server application to reverse the given input sentence.
5. Design TCP concurrent Client - Server application to reverse the given input sentence.
6. Design TCP Client - Server application to transfer file.
7. Design a TCP concurrent Server to convert a given text into upper case using multiplexing system call “select”.
8. Design a TCP concurrent Server to echo given set of sentences using Poll functions.
9. Design UDP Client - Server application to reverse the given input sentence.
10. Design UDP Client Server application to transfer a file.
11. Design using Poll Client Server application to multiplex TCP and UDP requests for converting a given text into upper case.
12. RPC application to add and subtract a given pair of integers.

TEXT BOOKS:

1. W R Stevens, “Unix Network Programming, Inter Process Communication”, 2nd edition, Prentice Hall of India/Pearson, 2014.
2. W R Stevens, “Unix Network Programming, the Sockets Networking API”, 3rd edition, Prentice Hall of India/Pearson, 2012.
3. Andrew S Tanenbaum, “Computer Networks”, 5th edition. Pearson Education, 2014.

REFERENCE BOOKS:

1. N Matthew, R Stones and Wrox, “Beginning Linux Programming”, 4th edition, Wiley India, 2008.
2. Robert Love, “Linux System Programming”, 2nd edition, O’Reilly, 2013.
3. Graham Glass and King Ables, “Unix for programmers and users”, 3rd edition, Pearson Education, 2003.

19CS337 CLOUD COMPUTING

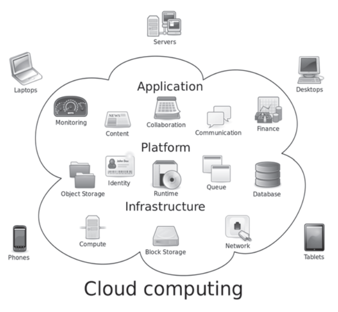
Hours Per Week :

L	T	P	C
3	0	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5



SOURCE: https://www.morlingglobal.in/

PREREQUISITE COURSES: Operating systems and Computer Networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course deals with computing model, which enables information, software, and shared resources to be provisioned over the network as services in an on-demand manner. The objective of this course is to enable the student to understand parallel and distributed computing, virtualization, architecture of cloud, aneka, task programming.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Analyze the trade-offs among deploying of applications in the cloud and the local infrastructure.	2
2	Evaluate the concepts of various virtualization technologies.	4
3	Deploy applications over commercial cloud computing infrastructures.	5
4	Identify security and privacy issues in cloud computing.	6, 8

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– I **L- 9**

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 – II **L- 9**

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.

UNIT – III **L- 9**

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.

UNIT – IV **L- 9**

CLOUD PLATFORMS IN INDUSTRY: Amazon web Services, Google app engine, Microsoft Azure.

CLOUD APPLICATIONS: Scientific applications in healthcare, biology, geo-science; Business applications in– CRM and ERP, productivity, social networking, media applications, multiplayer online gaming.

UNIT - V **L- 9**

ADVANCED TOPICS IN CLOUD COMPUTING: Energy efficiency in clouds, Market based management of clouds, Federated clouds / Inter Cloud, Third party cloud services.

LIST OF EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Launch Amazon Linux EC2 Instance and connect windows client to it.
2. Launch Windows EC2 instance in AWS and connect windows client to it..
3. Configure Web Server on Amazon Linux instance with Elastic IP.
4. Manage Elastic Block Storage(EBS).
5. Configure Amazon Simple Storage Service (Amazon s3).
6. Configure Amazon S3 Glacier.
7. Configure Amazon EFS.
8. Configure Amazon Virtual Private Cloud (VPC).

   a) Create your own VPC.

   b) Create public subnet.

   c) Create private subnet.

   d) Create a Internet gateway and attach to your VPC.

   e) Create Pubic 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.
9. Configure Amazon Elastic Load Balancer.
10. Configure Relational Database Service(RDS).

TEXT BOOK:

1. Raj Kumar Buyya, C Vecchiola and S TSelvi , “Mastering Cloud Computing”, 1st edition, Tata McGraw Hill Education (India), 2013.

REFERENCE BOOKS:

1. RajKumar Buyya, Broberg J and GoscinskiA, “Cloud Computing - Principles and Paradigms”, 1st edition, Wiley, 2011.
2. Rittinghouse J W, and Ransome J F, “Cloud Computing - Implementation, Management, and Security”, 1st edition, CRC Press, 2009.
3. Michael Wittig and Andreas Wittig, “Amazon Web Services in Action”, 2nd edition, Manning Publications, 2015.

source:
https://pdi.fbk.eu/

19CS338 ADVANCED DATA MINING

Hours Per Week :

L	T	P	C
3	0	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

COURSE DESCRIPTION AND OBJECTIVES:

This course enables the students with the advanced data mining techniques such as semi-supervised learning, web mining, social network mining and time series analysis. This course includes both theoretical and sill components. The theoretical part shares the methodological and algorithmic knowledge. Whereas the skill part encompasses with data analytics aids to solve real world problems.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Comprehend advanced Data Mining paradigms in semi-supervised learning.	1
2	Analyze the algorithmic constructs in web and sentiment mining.	2
3	Apply the advanced data mining techniques for crucial decision making.	1
4	Investigate the extracted patterns for web usage mining.	4
5	Evaluate the performance of data mining models on various kinds of data.	4

SKILLS:

- ✓ *Pre process the given data.*
- ✓ *Find the correlation among the attributes.*
- ✓ *Apply classification, association rule mining and clustering algorithms on data.*
- ✓ *Evaluate the classification and clustering methods performance.*

UNIT– I **L- 9**

SEMI-SUPERVISED LEARNING: Learning from labeled and unlabeled Examples, EM algorithm with naïve bayesian classification; Co-training; Self-training; Transductive support Vector Machines; Graph-based methods; Learning from positive and unlabeled examples, Applications of PU learning, Theoretical foundation; Building classifiers-two-step approach; Building classifiers; Biased-SVM, Building classifiers: Probability estimation.

UNIT – II **L- 10**

WEB MINING: Text pre-processing, Basic concepts of information retrieval, Information retrieval models; Relevance feedback, Evaluation measures, Text and Web page pre-processing, Inverted Index and Its compression, Latent Semantic indexing, Web Search, Meta-Search, Web Spamming.

UNIT – III **L- 8**

SOCIAL NETWORK ANALYSIS AND OPINION, SENTIMENT MINING: Social Network Analysis, Co-citation and bibliographic coupling, PageRank, HITS, Community discovery, Opinion mining and sentiment analysis, The problem of opinion mining, Document sentiment classification, Sentence subjectivity and sentiment classification, Aspect-based opinion mining.

UNIT – IV **L- 8**

WEB USAGE MINING: Web usage mining - data collection and pre-processing; Data modeling for Web usage Client - Server mining; Discovery and analysis of web usage patterns; Recommender systems and collaborative filtering; Query log mining.

UNIT - V **L- 10**

TIME SERIES ANALYSIS: Time series regression and exploratory data analysis, ARIMA models, Spectral analysis and filtering.

LIST OF EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

The Students pursue the following experiments by using the open source analytical tools such as R, Python, Weka, Rapid Miner etc. The following experiments enable the students to gain expertise in the fields of Natural Language Processing, Web Mining and Time Series Mining. Students experiment the following on UCI/ Kaggle/ NCBI data repository.

1) Classification on the following datasets with limited labels.

   a) Frauds and faults. b) Health care data.

2) Perform POS tagging, Named entity recognition (NER) and text summarization on news articles data.

3) Develop a recommender system to detect top K Communities in a social network community data.

4) Perform sentiment analysis/ opinion mining on twitter Data.

5) Demonstration of intrusion detection.

6) Extract hidden patterns by analyzing ECG and EEG datasets.

7) Develop a recommender system for the following:

   a) Finding of top trending movies b) Analysis of health care data.

8) Prediction on growth of a share value in a Stack exchange data.

TEXT BOOKS:

1. Bing Liu, "Web Data Mining", 2nd edition, Spinger, 2011.
2. Robert H. Shumway and David S. Stoffer, "Time series analysis and its applications with R examples", 4th edition, Springer, 2011.

REFERENCE BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining: Concepts and Techniques", 3rd edition, Morgan Kaufmann Publishers, 2011.
2. Bing Liu, "Sentiment analysis and opinion mining", 2nd edition, Morgan & Claypool Publishers, 2012,
3. Jure Leskovec, Anand Raja Raman and Jeffrey D Ullman, "Mining of Massive Datasets", 5th edition, Stanford University, 2014.

19CS431 INTERNET OF THINGS

SOURCE: https://taazaa.com/the-technologies-that-enable-the-internet-of-things/

Hours Per Week :

L	T	P	C
2	0	2	3

Total Hours :

L	T	P
30	-	30

WA/RA	SSH/HSH	CS	SA	S	BS
5	40	5	20	5	2

COURSE DESCRIPTION AND OBJECTIVES:

Students will be explored to the concepts and applications of Internet of Things, interconnection and integration of the physical world and the cyberspace. They are also able to design & develop IOT Devices and applications.

COURSE OUTCOMES:

Upon completion of the course, student will able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Analyze the application areas of IOT.	2
2	Realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks.	3
3	Analyze the building blocks of Internet of Things and characteristics.	2
4	Design and develop IoT applications for a given specific problem.	4

SKILLS:

- ✓ *Case studies on IoT applications.*
- ✓ *Write Python scripts for IoT circuits.*
- ✓ *Utilize the general purpose pins of suitable IoT.*

UNIT- I **L-6**

INTRODUCTION & CONCEPTS: Introduction to Internet of Things, Physical Design of IOT, Logical Design of IOT, IOT Enabling Technologies, IOT Levels.

UNIT - II **L-6**

DOMAIN SPECIFIC IOTS: Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Life Style.

UNIT -II **L-6**

Installing Python, Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling, Date/ Time Operations, Classes, Python Packages

UNIT - IV **L-6**

M2M & SYSTEM MANAGEMENT WITH NETCONF-YANG: M2M, Difference between IOT and M2M, SDN and NFV for IOT, Software Defined Networking, Network Function Virtualization, Need for IOT Systems Management, Simple Network Management Protocol, Limitations of SNMP, Network Operator Requirements, NETCONF, YANG.

UNIT - V **L-6**

IOT DESIGN METHODOLOGY: Case study using weather monitoring, IOT Physical Devices & Endpoints, What is an IOT Device, Exemplary, Device, Board, Linux on Raspberry Pi, Interfaces and Programming of IOT Devices.

LIST OF EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. a. Demonstration and study of Raspberry Pi board, GPIO Pins and familiarity of various sensors.

   b. Demonstration and study of other Hardware board of IoT such as Arduino Uno and NodeMCU.
2. Design and Implementation of controlling LED using Python in Raspberry Pi board.
3. Design and Implementation of sensing light through LDR using Python in Raspberry Pi board.
4. Design and Implementation of controlling LED through switch using Python in Raspberry Pi board.
5. Design and Implementation to find obstacles through sensor using Python in Raspberry Pi board.
6. Design and Implementation of sensing and display temperature using Python in Raspberry Pi board.
7. Design and Implementation of detecting noise through microphone sensor using Python in Raspberry Pi board.
8. Design and Implementation of output devices through relay using Python in Raspberry Pi board.
9. Design and Implementation of vibration sensor using Python in Raspberry Pi board.
10. Design and Implementation of uploading sensor data into cloud using Python.

TEXT BOOK:

1. Vijay Madisetti and Arshdeep Bahga," Internet of Things: A Hands-On Approach", 1st edition, Orient Blackswan Pvt. Limited, 2014.

REFERENCE BOOKS:

1. Adrian McEwen, "Designing the Internet of Things", 1st edition, Wiley Publishers, 2013.
2. Daniel Kellmereit, "The Silent Intelligence: The Internet of Things",1st edition, DND Ventures LLC, 2013.

Source: https://api.intechopen.com/

19CS432 MOBILE AD-HOC NETWORKS

Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P
45	-	-

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSES: Computer Networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on major aspects of ad hoc networking, from design through performance issues to application requirements. It starts with the design issues and challenges associated with implementations of ad hoc network applications. This includes mobility, disconnections, and battery power consumption. The course provides a detailed treatment of proactive, reactive, and hybrid routing protocols in mobile wireless networks. It also covers the IEEE 802.11 Wireless LAN and discusses their characteristics and operations. Through activities, the course gives students hands-on experience in designing a mobile ad hoc network using the NS2 simulator.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	summarize the protocols used at the MAC layer and scheduling mechanisms to express the mathematical properties.	1
2	apply proactive and reactive routing algorithms to find optimal paths.	1
3	analyze types of routing protocols used for unicast and multicast routing.	2
4	compare the performance of various routing protocols in adhoc networks.	4
5	develop the network security solution and routing mechanism.	3

SKILLS:

- ✓ *Evaluate various routing protocols.*
- ✓ *Analyze the performance of MAC protocols for Ad-hoc networks.*
- ✓ *Analyze the performance of Network protocols for Ad-hoc networks.*

UNIT– I **L- 9**

INTRODUCTION: Introduction to ad-hoc networks - definition, characteristics, features, applications; Characteristics of wireless channel; Ad-hoc mobility models - indoor and outdoor models.

UNIT – II **L- 9**

MEDIUM ACCESS PROTOCOLS: MAC protocols - design issues, goals and classification; Contention based protocols – with reservation, without reservation; Scheduling algorithms; Protocols using directional antennas; IEEE standards - 802.11a, 802.11b, 802.11g, 802.15; HIPERLAN.

UNIT – III **L- 9**

NETWORK PROTOCOLS: Routing protocols - design issues, goals and classification; Proactive Vs reactive routing; Unicast routing algorithms; Multicast routing algorithms; Hybrid routing algorithm; Energy aware routing algorithm; Hierarchical routing; QoS aware routing.

UNIT – IV **L- 9**

END – END DELIVERY AND SECURITY: Transport layer - issues in designing, transport layer classification, adhoc transport protocols; Security issues in adhoc networks - issues and challenges, network security attacks; Secure routing protocols.

UNIT - V **L- 9**

CROSS LAYER DESIGN: Cross layer design - need for cross layer design, cross layer optimization; Parameter optimization techniques; Cross layer cautionary perspective; Integration of adhoc with mobile IP networks.

TEXT BOOKS:

1. C. Siva Ram Murthy and B. S. Manoj, “Ad hoc Wireless Networks Architecture and Protocols”, 2nd edition, Pearson Edition, 2007.
2. Charles E. Perkins, “Ad hoc Networking”, 1st edition, Addison – Wesley, 2000.

REFERENCE BOOKS:

1. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan stojmenovic, “Mobile ad-hoc networking”, Wiley-IEEE press, 1st edition, 2004.
2. Mohammad Ilyas, “The Handbook of Adhoc Wireless Networks”, 1st edition, CRC press, 2002.
3. T. Camp, J. Boleng and V. Davies “A Survey of Mobility Models for Ad Hoc Network Research, ” Wireless Commun. and Mobile Comp., Special Issue on Mobile Ad Hoc-Networking Research, Trends and Applications, vol. 2, no. 5, 2002, pp. 483–502.
4. A survey of integrating IP mobility protocols and Mobile Ad hoc networks, Fekri M.Abduljalil and Shrikant K. Bodhe, IEEE communication Survey and tutorials, v no.1, 2007.
5. V.T. Raisinhani and S.Iyer “Cross Layer Design Optimization in Wireless Protocol Stacks” Comp. Communication, Vol 27 no. 8, 2004.
6. V.T.Raisinhani and S.Iyer,”ÉCLAIR; An Efficient Cross-Layer Architecture for Wireless Protocol Stacks”, World Wireless cong., San francisco, CA, May 2004.

19CS433 BIG DATA & ANALYTICS

source:
https://www.smartdatacollective.com/

Hours Per Week :

L	T	P	C
3	-	2	3

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSES: Databases; Design and Analysis of Algorithms.

COURSE DESCRIPTION AND OBJECTIVES:

This course gives an overview of Big Data, i.e.storage, retrieval and processing of big data.The focus will be on the “technologies”, i.e., the tools/algorithms that are available for storage and processing of Big Data and a variety of “analytics”.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand Big Data and its analytics in the real world.	1
2	Use the Big Data frameworks like Hadoop and NOSQL to efficiently store and process Big Data to generate Analytics.	2
3	Design of Algorithms to solve Data Intensive problems using Map Reduce Paradigm.	3
4	Design and Implementation of Big Data Analytics using Pig and Spark to solve Data Intensive problems and to generate analytics.	4
5	Analyse Big Data using Hive.	5

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.*

UNIT– I **L- 9**

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 – II **L-9**

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 nesource negotiator), Interacting with hadoop ecosystem.

UNIT – III **L- 9**

INTRODUCTION TO MAPREDUCE PROGRAMMING: Introduction-mapper, reducer, combiner, partitioner, searching, sorting, compression, real time applications using mapreduce, combiner, artitioner, matrix multiplication using mapreduce and page rank algorithm using mapreduce.

UNIT – IV **L-9**

INTRODUCTION TO PIG: The anatomy of pig, Pig on hadoop, Pig philosophy, Usecase 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.

INTRODUCTION TO HIVE: Introduction to hive, Hive architecture, Hive data types, Hive file format, Hive query language (HQL),

UNIT - V **L- 9**

HIVE: Partitions and bucketing, RCFile Implementation, working with XML files, User-defined Function (UDF) in Hive, Pig versus Hive.

SPARK: Introduction, features of spark, components of spark, programming with Resilient Distributed Datasets (RDD).

LIST OF EXPERIMENTS

LIST OF EXPERIMENTS **Total hours: 30**

1. HDFS basic command-line file operations.
2. HDFS monitoring User Interface.
3. WordCount Map Reduce program using Hadoop.
4. Implementation of word count with combiner Map Reduce program.
5. Practice on Map Reduce monitoring User Interface.
6. Implementation of Sort operation using MapReduce.
7. MapReduce program to count the occurrence of similar words in a file by using partitioner.
8. Design MapReduce 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.
9. MapReduce program to find Dept wise salary.

   Empno EmpName Dept Salary.
10. Install and Run Pig then write Pig Latin scripts to sort, group, join, project and filter the data.
11. Implementation of Word count using Pig.
12. Creation of Database and tables using Hive query language.
13. Creation of partitions and buckets using Hive.
14. Practice of advanced features in Hive Query Language: RC File & XML data processing.
15. Implement of word count using spark RDDs.
16. Filter the log data using Spark RDDs.

TEXT BOOKS:

1. Seema Acharya, Subhashini Chellappan, “Big Data and Analytics”, 1st edition, Wiley, Publishers, 2015.
2. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, Cambridge, University Press, 2012.
3. Holden Karau, Andy Konwinski, Patrick Wendell & Matei Zaharia,”Learning Spark”, 1st edition, Oreilly, 2015

REFERENCE BOOKS:

1. Boris lublinsky, Kevin t. Smith, AlexeyYakubovich, “Professional Hadoop Solutions”, 1st edition, Wiley, 2015.
2. Chris Eaton, Dirkderoosetal, “Understanding Big data “, 1st edition, McGraw Hill, 2012.
3. Tom White, “HADOOP: The definitive Guide”, 1st edition, O Reilly 2012.
4. Vignesh Prajapati, “Big Data Analytics with R and Haoop”,1st edition, Packet Publishing, 2013.

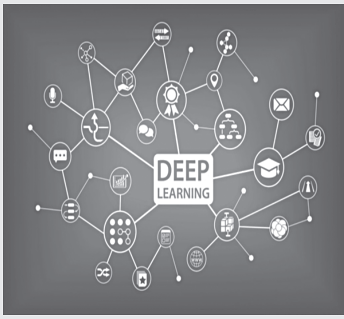
19CS434 DEEP LEARNING

Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P	CS	WA/RA	SSH	SA	S	BS
45	-	-	5	5	30	20	5	5



source:
https://cdn-images-1.medium.com/

PREREQUISITE COURSES: Machine learning; Probability & statistics.

COURSE DESCRIPTION AND OBJECTIVES:

This course offers sufficient details required to understand basic building blocks of various deep learning models. Especially, focuses on different types of neural network models like feed forward neural networks, convolutional neural networks, recurrent neural networks and auto encoders. During this course the students build, train and apply fully connected deep neural networks for various applications. In addition the students will learn how to apply hyper parameter tuning and best-practices to be followed for neural network training.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Design and implement the basic building blocks used in the Deep Learning based solutions.	1,3
2	Analyze and tune hyper parameters of a Deep Neural network model	2
3	Usage of tools to implement various deep learning models.	5
4	Application of Deep learning to solve various real-time problems.	6,9,10,12

SKILLS:

- ✓ *Develop strong foundations in Neural Networks and deep learning.*
- ✓ *Apply the techniques learned to solve the most common applications in Computer Vision and Natural Language Processing.*
- ✓ *Hyperparameter Tuning of a deep Neural network model.*
- ✓ *Neural network implementation in Tensor Flow/keras.*

UNIT– I **L- 9**

INTRODUCTION TO DEEP LEARNING: History of Deep Learning, What is Deep Learning?

FEEDFORWARD NEURAL NETWORKS: McCulloch–Pitts neuron, Perceptron learning rule, Perceptron convergence theorem, Sigmoidal neuron, Multi-layer feedforward neural network, backpropagation method, Gradient descent method, Stochastic gradient descent method.

UNIT – II **L- 10**

A**UTOENCODERS:** Autoassociative neural network, Stacked autoencoder, Autoencoders and relation to PCA, Regularization in autoencoders,Greedy layer-wisetraining, Variational auto encoder, Contractive autoencoder, Sparse autoencoder, Denoisingautoencoders.

UNIT – III **L- 8**

OPTIMIZATION AND REGULARIZATION METHODS FOR DEEP FFNN: Optimization methods - Adagrad, Adadelta, RMSProp, Adam; Regularization methods-Dropout, Dropconnect, Batch normalization; Activation functions - Linear, sigmoid, sigmoid, ReLU; Improving the training process - Early stopping, Parameter sharing and tying, Better weight initialization methods.

UNIT – IV **L- 8**

CONVOLUTIONAL NEURAL NETWORKS (CNNs): Basic CNN architecture, Advanced CNNs: LeNet, AlexNet, VGGNet, ResNet, Google Net and other architectures; 1-d CNN, 3-d CNN. Application of CNNs for computer vision tasks - image classification, video classification, object recognition and localization.

UNIT - V **L- 10**

SEQUENCE LEARNING: Architecture of an RNN, Unfolding of an RNN, Backpropagation through time, Long short term memory (LSTM), Gated recurrent units, Bidirectional RNNs and Bidirectional LSTMs.

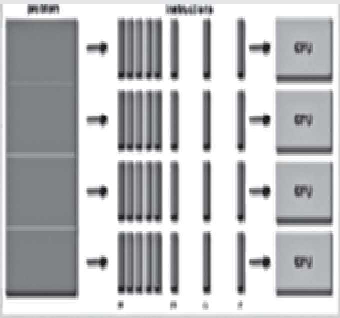
TEXT BOOKS:

1. Ian Goodfellow and Yoshua Bengio and Aaron, **“**Deep Learning**”,** 1st edition, An MIT Press Book, 2016

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.

19CS436 PARALLEL PROCESSING



source:
http://twimgs.com/ddj/

Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P
45	-	-

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSE: Computer Organization.

COURSE DESCRIPTION AND OBJECTIVES:

This course gives an overview of Parallel Computing, i.e.design and analyzethe parallel algorithms for real world problems and implement using Multicore processor systems.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.	2,3
2	Optimize the performance of a parallel program to suit a particular hardware and software environment.	3
3	Design algorithms suited for Multicore processor systems using OpenCL, OpenMP, Threading techniques.	2,3,5
4	Analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.	2
5	Understand the issues and design complexity for sorting techniques and graph algorithms using Parallel Processing.	3

SKILLS:

- ✓ *Understand the parallel computing and parallel programming platforms.*
- ✓ *Design parallel programs using Message Passing Interface.*
- ✓ *Design and build applications using POSIX threads.*
- ✓ *Learn tips and tricks to design parallel algorithms like sorting, graph algorithms etc.*

UNIT– I **L- 9**

INTRODUCTION TO PARALLEL COMPUTING: Motivating parallelism; Scope of parallel computing.

PARALLEL PROGRAMMING PLATFORMS: Implicit parallelism, Trends in microprocessor Architectures, Limitations of memory system performance, Dichotomy of parallel computing platforms, Physical organization of parallel platforms, Communication costs in parallel machines, Routing mechanisms for interconnection Networks, Impact of process-processor mapping and mapping techniques.

UNIT – II **L- 10**

PRINCIPLES OF PARALLEL ALGORITHM DESIGN: Preliminaries, Decomposition techniques, Characteristics of tasks and interactions, Mapping techniques for load balancing, Methods for containing interaction overheads, Parallel algorithm models.

BASIC COMMUNICATION OPERATIONS: One-to-All broadcast and All-to-One Reduction, All-to-All broadcast and reduction, All-Reduce and Prefix-Sum operations, Scatter and gather, All-to-All personalized communication, Circular Shift, Improving the speed of some communication operations.

UNIT – III **L- 8**

PROGRAMMING USING THE MESSAGE- PASSING PARADIGM: Principles of message-passing programming, The building blocks - send and receive operations; MPI - the message-passing interface, topologies and embedding; Overlapping communication with computation; Collective communication and computation operations, Groups and communicators.

UNIT – IV **L- 8**

PROGRAMMING SHARED ADDRESS SPACE PLATFORMS : Thread basics - why threads, the POSIX thread API, creation and termination, synchronization primitives in Pthreads, controlling thread and synchronization attributes; Thread cancellation; Composite synchronization constructs; Tips for designing asynchronous programs; OpenMP - a standard for directive based parallel programming.

UNIT - V **L- 10**

DENSE MATRIX ALGORITHMS: Matrix-vector multiplication; Matrix-matrix multiplication; Solving a system of linear equations.

SORTING: Issues in sorting on parallel computers; Sorting networks; Bubble sort and its variants; Quicksort; Bucket and sample sort; Other sorting algorithms.

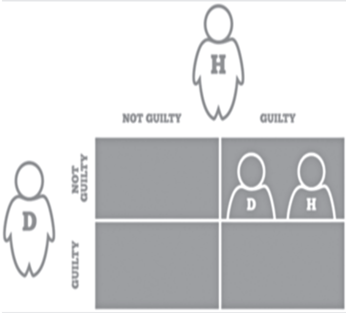
TEXT BOOKS:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "Introduction to Parallel Computing", 2nd edition Pearson Education –2007.
2. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill 1st edition, Computer Science Series, 2004.

REFERENCE BOOKS:

1. K. Hwang and F. A. Briggs, "Computer architecture and parallel processing", 1st edition, McGraw Hill Inc., 1985.
2. S. Akl., "Design and analysis of parallel algorithms", 1st edition, Prentice Hall Inc, 1992.

19CS437 GAME THEORY



source:
https://i.investopedia.com/

Hours Per Week :

L	T	P	C
3	0	0	3

Total Hours :

L	T	P
45	-	-

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSE: Computer Organization.

COURSE DESCRIPTION AND OBJECTIVES:

The objective of the course is to provide an intermediate course on Game Theory. Game theory is the science of strategy. It attempts to determine mathematically and logically the actions that “players” would take to secure the best outcomes for themselves in a wide array of “games.” Game Theory can be used to analyze the possible outcomes of situations ranging from card games and sports to strategic price fixing, negotiation, group cooperation. Game theory aids in understanding the possible advantage of moving first, the credibility of threats, the strategic importance of having a last encounter, and the mechanisms to maintain cooperation alive. Students will learn to recognize strategic environments and to use Game Theory to gain a better understanding of interactions and outcomes within them.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Identify strategic situations and represent them as games.	1
2	Solve simple games using various techniques.	2
3	Analyse economic situtations using game theoretic techniques.	2
4	Recommend and prescribe which strategies to implement.	3, 4

SKILLS:

- ✓ *Perfect and correct-information of Games.*
- ✓ *To know the relationship between backward induction and other solution.*
- ✓ *Computing the mixed-strategy Nash equilibria.*
- ✓ *Perfect-information of games with two or more players.*

UNIT– I **L- 9**

INTRODUCTION TO GAME THEORY: What is game theory?, Theory of rational choice, Interacting decision makers, Examples.

UNIT – II **L- 10**

STRATEGIC FORM SOLUTION CONCEPTS: Dominant strategy equilibrium; Dominance solvability; Strategic games - examples; Nash equilibrium - concept and examples; Best response functions, Dominated Actions, Symmetric games and symmetric equilibria.

UNIT – III **L- 8**

MIXED STRATEGY EQUILIBRIUM: Introduction, Mixed Strategies and expected Payoffs, Mixed strategy equilibrium, Dominated actions and mixed strategies.

UNIT – IV **L- 8**

BAYESIAN GAMES: Preliminaries, Bayesian equilibrium, some examples.

UNIT - V **L- 10**

EXTENSIVE FORM GAMES WITH PERFECT INFORMATION: Game trees, Strategies, Backward induction equilibrium, Commitment and mover advantages, Game Trees - a more formal treatment, extensive form Games with imperfect information.

TEXT BOOKS:

1. Levent Kockesen, “An Introduction to Game Theory”, New York University, 2007.
2. Martin J. Osborne and Ariel Rubinstein, “A Course in Game Theory” The MIT Press Cambridge, Massachusetts London, England, 2016.
3. Leyton-Brown K. and Shoham, Y., “Essentials of Game Theory: A Concise, Multidisciplinary Introduction”, Morgan & Claypool Publishers. 2008.
4. Osborne, M.J., “Introduction to Game Theory”, Oxford Univ. Press. 2004.

REFERENCE BOOKS:

1. Myerson, R.B., “Game Theory: Analysis of Conflict”, Harvard Univ. Press. 1991.
2. Watson, J., “Strategy: An Introduction to Game Theory”, Norton & Co., 2008.
3. Binmore, K., Playing for Real, “ A Text on Game Theory”,Course pack Edition: Oxford University Press. 2012.
4. Camerer, C.F., “Behavioral Game Theory: Experiments in Strategic Interaction”, Princeton University Press. 2003.

19CS438 HIGH PERFORMANCE COMPUTING

source: https://www.eletimes.com/wp-content/

Hours Per Week :

L	T	P	C
3	0	0	3

Total Hours :

L	T	P	CS	WA/RA	SSH	SA	S	BS
45	-	-	5	5	30	20	5	5

PREREQUISITE COURSES: Computer Organization.

COURSE DESCRIPTION AND OBJECTIVES:

The objective of the course is to introduce the fundamentals of high performance computing with the parallel processing units and many integrated cores using their architectures and corresponding programming environments. It provides a systematic and comprehensive treatment of the components in the pipeline that extract instruction level parallelism. And also provides a strong foundation on memory hierarchy design and tradeoffs in both uniprocessor and multiprocessors.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Implement high performance versions of standard single threaded algorithms.	1,2
2	Demonstrate the architec tural features in the OpenMP hardware accelerators.	2,3
3	Handle in a multicore, shared memory execution environment processor.	3
4	Design and deploy large scale parallel programs on tightly coupled parallel systems using the message passing paradigm.	2,3,4,5

SKILLS:

- ✓ *Understand the parallel computing and parallel programming platforms.*
- ✓ *Design parallel programs using Message Passing Interface.*
- ✓ *Design and build applications using POSIX threads.*
- ✓ *Learn tips and tricks to design parallel algorithms, like sorting, graph algorithms etc.*

UNIT– I **L- 8**

MODERN PROCESSORS: Stored program computer architecture; General purpose cache based microprocessor- performance metrics and benchmarks, Moore's Law, pipelining, superscalarity, SIMD; Memory hierarchies- cache, cache mapping, prefetch; Multicore processors; Multithreaded processors; Vector Processors- design principles, maximum performance estimatesl; Programming for vector architecture.

UNIT – II **L- 12**

BASIC OPTIMIZATION TECHNIQUES FOR SERIAL CODE : Scalar profiling-function and line based runtime profiling, hardware performance counters; Common sense optimizations; Simple measures, Large impact- elimination of common sub expressions, avoiding branches, using SIMD instruction sets;The role of compilers - general optimization options, inlining, aliasing, computational accuracy, register optimizations, using compiler logs; C++ optimizations – temporaries, dynamic memory management, loop kernels and iterators; Data access optimization- balance analysis and light speed estimates; Storage order; Case study - Jacobi algorithm and dense matrix transpose.

UNIT – III **L- 9**

PARALLEL COMPUTERS : Taxonomy of parallel computing paradigms; Shared memory computers-cache coherence, UMA, ccNUMA; Distributed memory computers; Hierarchical systems; Networks-basic performance characteristics, buses, switched and fat tree networks, mesh networks, Hybrids; Basics of parallelization - why parallelize, data parallelism, function parallelism; Parallel scalability-factors that limit parallel execution, scalability metrics, simple scalability laws, parallel efficiency, serial performance vs strong scalability, refined performance models,choosing the right scaling baseline; Case study - can slow processors compute faster, load imbalance.

UNIT – IV **L- 8**

DISTRIBUTED MEMORY PARALLEL PROGRAMMING WITH MPI: Message passing, Introduction to MPI –example, messages and point-to- point communication, collective communication, nonblocking point-to-point communication, virtual topologies, MPI parallelization of Jacobi solver, MPI implementation, performance properties.

UNIT - V **L- 8**

SHARED MEMORY PARALLEL PROGRAMMING WITH OPENMP: Shared memory parallel programming with OpenMp; Introduction to OpenMp - parallel execution, data scoping, OpenMp work sharing for loops, synchronization, reductions, loop scheduling, tasking; case study: OpenMp-parallel jacobi algorithm, advanced OpenMp wave front parallelization; Efficient OpenMp programming - profiling OpenMp programs, performance pitfall; Case study- parallel sparse matrix, vector multiply.

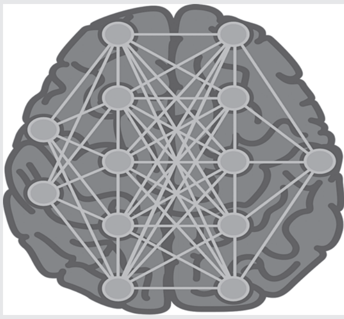
TEXT BOOKS:

1. Georg Hager and Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers", Chapman & Hall/CRC Computational Science series, 2011.
2. AnanthGrama, Anshul Gupta, George Karypis and Vipin Kumar: "Introduction to Parallel Computing", 2nd edition Pearson Education, 2007.

REFERENCE BOOKS:

1. Charles Severance and Kevin Dowd, "High Performance Computing", O'Reilly Media, 2nd edition, 1998.
2. Kai Hwang and Faye Alaye Briggs, "Computer Architecture and Parallel Processing", McGraw Hill, 1984.

19CS439 ARTIFICIAL NEURAL NETWORKS



source: *https://ds055uzetaobb.cloudfront.net/brioche.*

Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P	CS	WA/RA	SSH	SA	S	BS
45	-	-	5	5	30	20	5	5

PREREQUISITE COURSES: Engineering Mathematics; Probability & Statistics.

COURSE DESCRIPTION AND OBJECTIVES:

The course introduces the principles of neuro-computing with artificial neural networks, which widely used for addressing real-world problems such as classification, regression, pattern recognition, data mining, time-series modelling, etc.. This course majorly covers two kinds of learning such as supervised and unsupervised.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand the differences between networks for supervised and unsupervised learning.	1
2	Apply the linear and nonlinear models for learning the data.	1
3	Analyse the performance of various neural networks on different kinds of data.	3
4	Evaluate the neural networks for classify/ cluster the data to achieve higher performance.	4
5	Design/ Develop different neural networks such as MLP, SOM, Hopfiled net and ART etc.	5

SKILLS:

- ✓ *Learn to design and build different neural network models.*
- ✓ *Learn to develop learning algorithms for both supervised/ unsupervised learning.*
- ✓ *Gain the knowledge for network tuning, generalization and address over-fitting problem.*

UNIT– I **L- 9**

INTRODUCTION TO ARTIFICAL NEURAL NETWORKS: Introduction; Artificial neural networks; Historical development of neural networks; Biological Neural Networks; Comparison between them and the Computer; Comparison between artificial and biological neural network; Basic building blocks; Terminologies.

UNIT – II **L- 9**

FUNDAMENTAL MODELS OF ARTIFICIAL NEURAL NETWORKS: Introduction, McCulloch-Pitts Neuron model; Learning rules - Hebbian learning rule, perceptron learning rule, Delta learning rule, Widrow-Hoff rule or least mean square (LMS) rule, Competitive learning rule; Out star learning; Boltzmann based learning; Hebb net.

PERCEPTRON NETWORKS : Introduction; Single layer perceptron; Brief introduction to multilayer perceptron networks.

UNIT – III **L- 9**

ADALINE AND MADALINE NETWORKS: Introduction, Adaline, Madaline.

ASSOCIATIVE MEMORY NETWORKS: Introduction, Algorithms for pattern association, Hetero associative memory neural networks, Auto associative memory network, Bi-directional associative memory.

UNIT – IV **L- 9**

FEEDBACK NETWORKS: Introduction, Discrete Hopfiled Net, Continuous Hopfiled Net, Relation between BAM and Hopfiled Nets.

FEED FORWARD NETWORKS: Introduction, Back Propagation Network (BPN), Radial Basis Function Network (RBFN).

UNIT - V **L- 9**

SELF ORGANIZING FEATURE MAP: Introduction; Methods used for determining the winner; Kohonen self organizing feature maps; Learning vector quantization(LVQ); MaxNet, Maxican Hat, Hamming Net

ADAPTIVE RESONANCE THEORY: Introduction, ART Fundamentals, ART1, ART2.

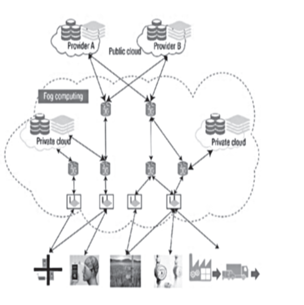
TEXT BOOK:

1. Sivanandam, S.Sumathi and S.N.Deepa; “Introduction toNeural Networks”, 2nd edition., TATA McGraw HILL: 2005.

REFERENCE BOOKS:

1. Simon. Hhaykin, “Neural networks A comprehensive foundations”, 2nd edition, Pearson Education, 2004.
2. B. Yegnanarayana, “Artificial neural networks”, 1st edition., Prentice Hall of India Pvt. Ltd, 2005.
3. Li Min Fu, “Neural networks in Computer intelligence”,1stedition., TMH, 2003.

19CS440 DISTRIBUTED SYSTEMS



source:
https://www.icar.cnr.it/

Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P
45	-	-

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSE: Computer Networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on understanding the foundations of distributed systems and detailed virtualization of networks such as RMI (Remote Method Invocation which is required for distributed systems. It also provides the idea of peer to peer services and file system. In addition, it gives the knowledge about clock synchronization techniques, transactions, concurrency control mechanisms, issues in process and resource management.At the end students will be familiar with the design, implementation and security issues of distributed system.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Apply the knowledge of distributed systems techniques, trends and methodologies.	1
2	Learn and apply the concept of network virtualization and remote method invocation.	1
3	Analyze the mechanism of peer to peer systems, DFS and DNS.	3
4	Understand key mechanisms and models for distributed systems including logical, clocks, causality, distributed mutual exclusion, distributed deadlocks.	4
5	Gain experience in learning process and resource management.	5

SKILLS:

- ✓ *Compare replication schemes with respect to performance, availability and consistency concerns.*
- ✓ *Design, implement, and debug distributed systems.*

UNIT– I **L- 9**

INTRODUCTION: Introduction of Distributed Systems, Trends in Distributed Systems, Focus on resource sharing, Challenges, Case study - WWW.

UNIT – II **L- 10**

COMMUNICATION IN DISTRIBUTED SYSTEM: System model-inter process communication, the API for internet protocols; External data representation and Multi-cast communication; Network virtualization; Overlay networks; Case study - MPI remote method invocation and objects; Remote invocation – introduction, request-reply protocols, remote procedure call, remote method invocation; Case study - Java RMI; Group communication - publish-subscribe systems, message queues, shared memory approaches, distributed objects; Case study - enterprise Java Beans from objects to components.

UNIT – III **L- 10**

PEER TO PEER SERVICES AND FILE SYSTEM: Peer-to-peer Systems, Introduction, Napster and its legacy, Peer-to-peer, middleware, routing overlays; Overlay case studies - pastry, tapestry; Distributed file systems – introduction, file service architecture, Andrew file system; File system - features, file model, file accessing models; File sharing semantics naming - identifiers, addresses, name resolution, name space implementation, name caches, LDAP.

UNIT – IV **L- 9**

SYNCHRONIZATION AND REPLICATION: Introduction - clocks, events and process states, synchronizing physical clocks, logical time and logical clocks, global states; Coordination and agreement – introduction, distributed mutual exclusion, elections; Transactions and concurrency control– transactions, nested transactions, locks, optimistic concurrency control, time stamp ordering, atomic commit protocols; Distributed deadlocks – replication; Case study – CODA.

UNIT - V **L- 9**

PROCESS & RESOURCE MANAGEMENT: Process management; Process migration - features, mechanism; Threads - models, issues, implementation; Resource management - introduction, features of scheduling algorithms, task assignment approach, load balancing approach, load sharing approach.

TEXT BOOK:

1. George Coulouris, Jean Dollimore and Tim Kindberg, “Distributed Systems Concepts and Design”, 5th edition, Pearson Education, 2012.

REFERENCE BOOKS:

1. Pradeep K Sinha, “Distributed Operating Systems: Concepts and Design”, 2nd edition, Prentice Hall of India, 2007.
2. Tanenbaum A.S and Van Steen M., “Distributed Systems: Principles and Paradigms”, 2nd edition, Pearson Education, 2007.
3. Liu M.L., “Distributed Computing, Principles and Applications”, 5th edition, Pearson Education, 2004.
4. Nancy A. Lynch, “Distributed Algorithms”, 4th edition,Morgan Kaufman Publishers, USA, 2013.

19CS441 WIRELESS SENSOR NETWORKS

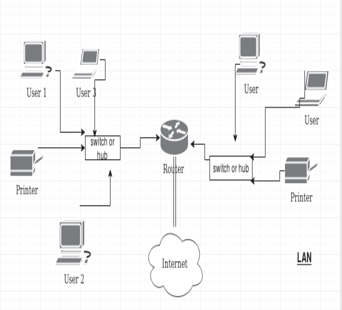
Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P
45	-	-

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5



Source https:// cdn.cnn.com/

PREREQUISITE COURSES: Computer Networks; MANET.

COURSE DESCRIPTION AND OBJECTIVES:

This course provides students with an opportunity to learn the fundamentals behind the design of wireless sensor networks. A primary focus of this course is to give students hands-on programming experience with various sensors and sensing platforms.

COURSE OUTCOMES:

Upon completion of the course, student will able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Apply the different routing protocol with respect to the WSN.	1
2	Analyze the difference between conventional Operating System Vs. Embedded Operating Systems.	2
3	Analyze the problem specific Medium Access Control Protocol.	2
4	Compare and differentiate between the WSN and Ad-hoc Networks.	4
5	Develop the WSN Applications with respect any domain specific requirements.	3, 6

SKILLS:

- ✓ *Establish local area networks with different topologies.*
- ✓ *Evaluate various routing protocols.*
- ✓ *Apply knowledge on different sensors and design of different technology circuits using sensors.*
- ✓ *Utilize a system approach to design a chip and operational performance.*