22TP103 PROGRAMMING IN C

Source : https://www.gramedia.com/pendidikan/jurusan-bioentrepreneurship/

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.

Practice Questions on Data Handling – Level 1:

1. Write a program to accept a character as input from the user and print it.
2. Write a program to accept a number as input from the user and print it.
3. Write a program to accept a float value from the user and print it.
4. Write a program to accept a message as input from the user and print it.
5. Write a program to accept a message from the user as input and print it in 3 different lines.
6. Write a program to accept 2 numbers from the user as input and print their sum.
7. Write a program to accept 2 numbers from the user as input and print their product.
8. 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.

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.*

9. Write a program to accept a number as input from the user which denotes the radius and print the area of the circle.
10. Write a program to accept a character as input from the user and print it's corresponding ASCII value.

Questions on Control Statements - Looping – Level 1:

1. Write a C program to print all the characters from a to z once.
2. Write a C program to print all the characters from Z to A once.
3. Write a C program to print all the characters from A to Z 3 times.
4. Write a C program to print the first N natural numbers, where N is given as input by the user.
5. Write a C program to print the first N natural numbers and their sum, where N is given as input by the user.
6. Write a C program to print all the odd numbers between 1 and N where N is given as input by the user.
7. Write a C program to print all the even numbers between I and N where N is given as input by the user.
8. 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.
9. 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.
10. 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:

1. Write a program to accept two numbers as input check if they are equal.
2. Write a program to accept two characters as input and check if they are equal.
3. Write a program to accept two numbers as input and print the greater of the 2 numbers.
4. Write a program to accept two numbers as input and print the lesser of the 2 numbers.
5. Write a program to accept 3 numbers as input and print the maximum of the 3.
6. Write a program to accept 3 numbers as input and print the minimum of the 3.
7. 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.
8. 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.
9. 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.
10. 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.

```
*****
*   *
*   *
*   *
*****
```

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:

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

Questions on Arrays – Level 1:

1. Print the contents of an array from the left to the right.
2. Print the contents of an array from the right to the left.
3. Find the sum of the elements of an array.
4. Find the maximum element in an unsorted array.
5. Find the minimum element in an unsorted array.
6. Find the average of the elements in an unsorted array.
7. Count the number of 0s and 1s in an array having 0s and 1s in random order.
8. Count the number of elements in an array whose elements are lesser than a key element in an unsorted array.
9. Print all the elements in an array whose values are lesser than a key element in an unsorted array.
10. Find the repeated elements in a sorted array.

Questions Number crunching – Level 2:

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

Questions on Arrays – Level 2:

1. Find the sum of the maximum and minimum numbers of an unsorted array.
2. Replace every element in an array with the sum of its every other element.
3. Replace every element in an array with the sum of its right side elements.
4. Replace every element in an array with the sum of its left side elements.
5. Reverse the elements of an array (in place replacement).
6. Reverse the first half of an array.
7. Reverse the second half of an array.
8. Write a program to find the second largest element in an unsorted array.
9. Write a program to find the second smallest element in an unsorted array.
10. Write a program to print the number of odd and even numbers in an unsorted array.

Questions on Strings – Level 1:

1. Write a program to accept a string as input and print it.
2. Write a program to accept a string as input and count the number of vowels in it.
3. Write a program to accept a string as input and count the number of consonants in it.
4. Write a program to accept a string as input and print its length.
5. Write a program to accept a string as input and print the reversed string.
6. Write a program to accept 2 strings as input and check if they are the same.
7. Write a program to accept a string as input and copy the contents into a second string and print the second string.
8. Write a program to accept 2 strings as input and concatenate them into a third string and print the third string.
9. Write a program to accept a string as input and check if it is a palindrome.
10. 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:

1. Implement the string length function.
2. Implement the string copy function.
3. Implement the string concatenate function.
4. Implement the string compare function.
5. Implement the vowel count function.
6. Implement the consonant count function.
7. Implement the count words function.
8. Implement the string reverse function.
9. Implement the strstr function.
10. 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:

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

Questions on 2D Arrays – Level 1:

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

Questions on 2D Arrays – Level 2:

1. Find and print the maximum element of each row of a matrix.
2. Find and print the minimum elements of each row of a matrix.
3. Find and print the maximum element of each column of a matrix.
4. Find and print the minimum element of each column of a matrix.

5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. Write a program to search a given element in a row-wise and column-wise sorted 2D array.

Questions on 2D Arrays – Level 3:

1. Write a program to find the Kth smallest element in the given matrix.
2. Write a program to find the Kth largest element in the given matrix.
3. Write a program to check whether the given two two-dimensional array of same dimensions are equal or not.
4. Write a program to add the given two two-dimensional array of same dimensions.
5. Write a program to subtract the given two two-dimensional array of same dimensions.
6. Write a program to multiply the given two two-dimensional array of same dimensions.
7. Write a program to sort each row of a matrix.
8. Write a program to find the sum of the elements in 'Z' sequence of the given 2D array.
9. Write a program to print the unique rows of the given two-dimensional array consisting of only 0's and 1's.
10. 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.

   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.

6. 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.

7. 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
```

8. 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
   ```

9. 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.

10. 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.

11. 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.”

12. 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

13. 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 the 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:

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.