PROBLEM SOLVING AND PROGRAMMING WITH C

L	T	P	C
0	0	6	3

The course is designed to enable the student to write programs for problem solving. After an introduction to program logic design using algorithms and flowcharts, converting the logic into programs is taught. The features of structured programming are explained with the C programming language as an example. This course lays the foundation both for developing program logic and for writing programs in C according to the developed logic.

Course objectives:

1.Familiarize the student with the steps involved in writing and running a compiled program.

2.Enable the student to build program logic with algorithms and flowcharts.

3. Explain with the features and constructs of C programming such as data types, expressions, loops, functions, arrays, pointers, and files.

4.Demonstrate the handling of variables and input-output operations in C.

5.Train the student to convert program logic into C language code using a top-down approach.

Module I:	Introduction to Computer Problem-Solving	12	P

Introduction, the Problem-Solving Aspect, Top-Down Design, Introduction to the idea of an algorithm, Introduction to Flowchart using Raptor tool.

Introduction to C Language – Structure of a C Program, Keywords, Identifiers, Data Types (int, float, char, unsigned int) and Variable declaration, Constants, Input / Output function. Operators, Expressions, Precedence and Associativity, Expression Evaluation, Type conversions.

Exercises: Construct a flowchart and write a program to

- Develop a calculator to convert time, distance, area, volume and temperature from one unit to another.
- Calculate simple and compound interest for various parameters specified by the user
- To enter marks of five subjects and calculate total, average and percentage.
- Calculate net salary of employee given basic,da,hra,pf and lic
- retrieve remainder after division of two numbers without using mod operator
- Convert an upper-case character to a lower-case character.
- Swap two numbers
- Enter two angles of a triangle and find the third angle.
- Check Least Significant Bit (LSB) of a number
- Input any number from user and check whether nth bit of the given number is set (1) or not (0)(hint: Use bitwise operators)

After completion of this unit the student will be able to

- Develop algorithms and basic flowcharts for performing Input, Output and Computations (L3)
- Interpret the structure of C program and various key features of C (L2)
- Translate mathematical expressions to C notation using operators (L2).

Module II:	Control Structures	15	P

- **Control Structures:** Selection Statements (making decisions) – if, if-else, nested if, else if ladder and switch statements. Repetition statements (loops)-while, for, do-while statements, Nested Loops.
- Unconditional statements-break, continue, goto.
- Pointers – Pointer variable, pointer declaration, Initialization of pointer, accessing variables through pointers, pointers to pointers, pointers to void.

Exercises: Construct a Flowchart and Write a Program to

- Check whether the triangle is equilateral, isosceles, or scalene triangle.
- Check whether entered year is a leap year or not
- Find minimum among three numbers.
- Check whether a number is divisible by 5 and 11 or not.
- Check whether a number is positive, negative or zero using switch case.
- Design a calculator that performs arithmetic operations on two numbers using switch case
- Find Roots of a Quadratic Equation
- Find factorial of a number
- Check whether number is a palindrome or not
- Check whether number is perfect or not
- Convert a decimal number to binary number
- To find the sum of the series [1-X^2/2!+X^4/4!- ....].
- Print following patterns

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


A
B B
C CC
D D D D
E E E EE


1
2 3
4 5 6
7 8 9 10
```

- Calculate the greatest common divisor of two numbers
- Generate first n numbers in the Fibonacci series
- Generate n prime numbers
- Swap two numbers using pointers.
- Performs all the five arithmetic operations using Pointers.

Learning Outcomes:

After completion of this unit the student will be able to

- Construct C programs using various conditional statements (L3).
- Develop C programs using loops and nested loops (L6).
- Demonstrate the usage of pointers (L3).

Module III:	Functions	15	P

Functions-Designing Structured Programs, user defined function- function definition, function prototype, function call, Types of functions. Parameter Passing by value, parameter passing by address, Recursive functions. Dynamic Memory allocation Functions, pointers to functions.

Storage classes-auto, register, static, extern.

Exercises: Write a program using functions to

- Print even and odd numbers in a given range
- Find power of a number
- Return maximum of given two numbers
- To print all strong numbers between given interval using functions.
- Check whether a number is prime, Armstrong or perfect number using functions.
- Demonstrate call by value and call by reference mechanisms.
- Find power of any number using recursion.
- Generate Fibonacci series using recursion
- Find product of two numbers using recursion
- Find the sum of digits of a number. Number must be passed to a function using pointers.
- Find GCD (HCF) of two numbers using recursion.
- Find LCM of two numbers using recursion.

Learning Outcomes:

After completion of this unit the student will be able to

- understand the concept of subprograms and recursion (L2).
- apply the in-built functions to develop custom functions for solving problems (L3).
- make use of parameter passing mechanisms (L3).
- infer the effect of storage classes on variables (L2).

Module IV:	Arrays and Strings	15	P

Arrays – Declaration and Definition of Array, accessing elements in array, Storing values in array, linear search, binary search, bubble sort, Two – dimensional arrays, multidimensional arrays. Arrays and Pointers, Pointer Arithmetic and arrays, array of pointers, Passing array to function.

Strings – Declaration and Definition of String, String Initialization, unformatted I/O functions, arrays of strings, string manipulation functions, string and pointers.

Exercises: Write a program to

- Find minimum and maximum element in an array
- Implement linear search.
- Sort an array in descending order.
- Given a two-dimensional array of integers and a row index, return the largest element in that row.
- Find transpose of a matrix.
- Perform multiplication of two matrices
- Count total number of vowels and consonants in a string.
- Reverse the given string without using String handling functions.
- Sort strings in dictionary order
- To perform addition of two matrices.
- Read an array of elements of size 'n' and find the largest and smallest number using functions
- find total number of alphabets, digits or special character in a string using function

Learning Outcomes:	

After completion of this unit the student will be able to

- develop programs for storing and managing collections of items using arrays (L3).
- make use of the in-built functions to manipulate strings (L3).
- solve problems related to arrays and strings (L3).

Module V:	**Structures and Files**	15	P

Structures–Declaration, initialization, accessing structures, operations on structures, structures containing arrays, structures containing pointers, nested structures, self-referential structures, arrays of structures, structures and functions, structures and pointers, unions.
Files – Concept of a file, Opening and Closing files, file input / output functions (standard library input / output functions for text files)

Exercises: Write a program to

- Store information of a student using structure
- Add two complex numbers by passing structures to a function
- Store information of 10 students using structures
- Store Employee information using nested structure
- Read file contents and display on console.
- Read numbers from a file and write even and odd numbers to separate file.
- Count characters, words and lines in a text file.

Learning Outcomes:	

After completion of this unit, the student will be able to:

- develop programs using structures and unions for storing dissimilar data items (L6).
- compare the utilization of memory by structures and unions (L5).
- make use of files and file operations to store and retrieve data (L3).

Text Books(s)

- B. A. Forouzan and R. F. Gilberg, Computer Science: A Structured Programming Approach Using C, 3/e, Cengage Learning

Reference Book(s)

- Jeri R Hanly, Elliot B Koffman, Problem Solving and Program Design in C, 7/e, Pearson Education, 2012.
- B.W. Kernighan and Dennis M. Ritchie, The C Programming Language, 2/E, Pearson education, 2015.
- B. Gottfried, Programming with C, 3/e, Schaum's outlines, McGraw Hill (India), 2017.
- P. Dey and M Ghosh, Programming in C, 2/e, Oxford University Press, 2011.

Course Outcomes:

After completion of this course the student will be able to

- Build logic for solving a problem and translate it into a program. (L3).
- Define variables and construct expressions using C language (L1).
- Utilize arrays, structures and unions for storing and manipulating data (L3).
- Develop efficient, modular programs using functions (L3).
- Write programs to store and retrieve data using files (L3).

Additional Exercises:

- Given numbers x, y, and target, return whichever of x and y is closer to the target. If they have the same distance, return the smaller of the two
- There are three friends Ram, Raheem and Robert. Ram's age is 20, Raheem is aged three times more than his friend Ram. After 8 years, he would be two and a half times of Ram's age. After further 8 years, how many times would he be of Rams age? Robert's age is 25 now. Now program your computer to determine the final ages of all the three people after 16 years and also show who is elder.
- Given an actual time and an alarm clock time, both in "military" format (such as 0730 for 7:30am), print how many more minutes before the alarm rings. But if the time is after the alarm, print "Alarm already went off".
- Let there be a scenario where you and your friend are going to a restaurant. You have lunch there every fourth day, and he has his lunch there every sixth day. How many days before you meet again for lunch at the same restaurant?
- Two friends Suresh and Ramesh have **m** red candies and **n** green candies respectively. They want to arrange the candies in such a way that each row contains equal number of candies and also

each row should have only red candies or green candies. Help them to arrange the candies in such a way that there are maximum number of candies in each row.

- On a chessboard, positions are marked with a letter between a and h for the column and a number between 1 and 8 for the row. Given two position strings, return true if they have the same colour.
- Given two strings s0 and s1, return whether they are anagrams of each other.
- Write a program to encrypt and decrypt a password which is alphanumeric
- Given a string, return the string with the first and second half swapped. If the string has odd length, leave the middle character in place.
- Given an array of integers, return the second-largest element.
- Given lists of integers people, jobs, profits. Each person i in people have people[i] amount of strength, and performing job j requires jobs[j] amount of strength and nets profits[j] amount of profit. Given that each person can perform at most one job, although a job can be assigned to more than one person, return the maximum amount of profit that can be attained.
- Mr. Roxy has arranged a party at his house on the New Year’s Eve. He has invited all his friends - both men and women (men in more number). Your task is to generate the number of ways in which the invitees stand in a line so that no two women stand next to each other. Note that the number of men is more than the number of women and Roxy doesn’t invite more than 20 guests. If there are more than 20 guests or an arrangement as per the given constraints is not possible, print ‘invalid’.
- Two friends have entered their date of birth and they want to know who is elder among them. Make a structure named Date to store the elements day, month and year to store the dates.

Case Study:

- Create a structure containing book information like accession number, name of author, book title and flag to know whether book is issued or not. Create a menu in which the following functions can be done: Display book information, Add a new book, Display all the books in the library of a particular author, Display the number of books of a particular title, Display the total number of books in the library, Issue a book (If we issue a book, then its number gets decreased by 1 and if we add a book, its number gets increased by 1)
- Ranjan is maintaining a store. Whenever a customer purchases from the store, a bill is generated. Record the customer name, amount due, the amount paid, mobile number with purchased items in file. At the end of day print the total income generated by store.
- Contact Management System- Create structure to store Contact information like name,gender,mail,phone number and address. Users can add new contact and can also edit and delete existing contact. (Hint: Use Files to store data)