PROGRAMMING FOR PROBLEM SOLVING LAB

I B.TECH- Common to all Branches

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
A5CS02	ESC	-	-	3	1.5	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 48				Total Classes:48		

COURSE OBJECTIVES:

1. To be familiarize with flowgorithm to solve simple problems
2. To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
3. To develop modular, reusable and readable C Programs using the concepts like functions, arrays, strings, pointers and structures.

COURSE OUTCOMES

At the end of the course, student will be able to

1. Solve simple mathematical problems using Flowgorithm.
2. Correct syntax errors as reported by the compilers and logical errors encountered at run time
3. Develop programs by using decision making and looping constructs.
4. Implement real time applications using the concept of array, pointers, functions and structures.
5. Solve real world problems using matrices, searching and sorting

LIST OF EXPERIMENTS

Week - 1 INTRODUCTION TO FLOGORITHM

a. Installation and working of Flowgorithm Software.
b. Write and implement basic arithmetic operations using Flowgorithm – sum, average, product, difference, quotient and remainder of given numbers etc.

Week - 2 FLOWGORITHM - OPERATORS AND EVALUATION OF EXPRESSIONS

a. Draw a flowchart to calculate area of Shapes (Square, Rectangle, Circle and Triangle).
b. Draw a flowchart to find the sum of individual digits of a 3 digit number.
c. Draw a flowchart to convert days into years, weeks and days.
d. Draw a flowchart to read input name, marks of 5 subjects of a student and display the name of the student, the total marks scored, percentage scored.

Week - 3 FLOWGORITHM –CONDITIONAL STATEMENTS

a. Draw a flowchart to find roots of a quadratic equation.
b. Draw a flowchart to find the largest and smallest among three entered numbers and also display whether the identified largest/smallest number is even or odd
c. Draw a flowchart to check whether the triangle is equilateral, isosceles or scalene triangle

Week - 4 OPERATORS

a. Write a C program to swap values of two variables with and without using third variable.
b. Write a C program to enter temperature in Celsius and convert it into Fahrenheit.
c. Write a C program to calculate Simple and Compound Interest.
d. Write a C program to calculate s = ut+(1/2)at^2 where u and a are the initial velocity in m/sec (= 0) and acceleration in m/sec^2 (= 9.8 m/s^2)).

Week - 5	CONDITIONAL STATEMENTS

a. Write a C program to find largest and smallest of given numbers.
b. Write a C program which takes two integer operands and one operator form the user(+,-,*,/,% use switch)
c. Write a program to compute grade of students using if else adder. The grades are assigned as followed:

marks<50	F
50≤marks< 60	C
60≤marks<70	B
70≤marks	B+
80≤marks<90	A
90≤mars≤ 100	A+

Week - 6	LOOPING STATEMENTS

a. Write a C program to find Sum of individual digits of given integer
b. Write a C program to generate first n terms of Fibonacci series
c. Write a C program to generate prime numbers between 1 and n
d. Write a C Program to find the Sum of Series SUM=1-x2/2! +x4/4!-x6/6!+x8/8!-x10/10!
e. Write a C program to generate Pascal's triangle.
f. Write a C program to generate pyramid of numbers.

```
          1
      1   3   1
  1   3   5   3   1
```

Week - 7	ARRAYS

a. Write a C Program to implement following searching methods
   i. Binary Search
   ii. Linear Search
b. Write a C program to find largest and smallest number in a list of integers
c. Write a C program
   i. To add two matrices
   ii. To multiply two matrices
d. Write a C program to find Transpose of a given matrix

Week - 8	FUNCTIONS

a. Write a C program to find the factorial of a given integer using functions
b. Write a C program to find GCD of given integers using functions
c. Write a C Program to find the power of a given number using functions

Week - 9	RECURSION

a. Write a C Program to find binary equivalent of a given decimal number using recursive functions.
b. Write a C Program to print Fibonacci sequence using recursive functions.
c. Write a C Program to find LCM of 3 given numbers using recursive functions

Week - 10	STRINGS

a. Write a C program using functions to
   a. Insert a sub string into a given main string from a given position
   b. Delete n characters from a given position in a string
b. Write a C program to determine if given string is palindrome or not

Week - 11	POINTERS

a. Write a C program to print 2-D array using pointers
b. Write a C program to allocate memory dynamically using memory allocation functions (malloc, calloc, realloc, free)

Week - 12	STRUCTURES

a. Write a C Program using functions to
   i. Reading a complex number
   ii. Writing a complex number
   iii. Add two complex numbers
   iv. Multiply two complex numbers

   Note: represent complex number using structure

b. Write a C program to read employee details employee number, employee name, basic salary, hra and da of n employees using structures and print employee number, employee name and gross salary of n employees.

TEXT BOOKS:

1. Riley DD, Hunt K.A. Computational Thinking for the Modern Problem Solver. CRC press, 2014 Mar 27.
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)
3. Yashavant Kanetkar, "Let Us C", BPB Publications, New Delhi, 13th Edition, 2012.

REFERENCE BOOKS:

1. Ferragina P, Luccio F. Computational Thinking: First Algorithms, Then Code. Springer; 2018
2. King KN, "C Programming: A Modern Approach", Atlantic Publishers, 2nd Edition, 2015.
3. Kochan Stephen G, "Programming in C: A Complete Introduction to the C Programming Language", Sam's Publishers, 3rd Edition, 2004.
4. Linden Peter V, "Expert C Programming: Deep C Secrets", Pearson India, 1st Edition, 1994.

WEB REFERENCES:

1. http://www.flowgorithm.org/documentation/
2. http://www.sanfoundry.com/c-programming-examples
3. http://www.geeksforgeeks.org/c
4. http://www.cprogramming.com/tutorial/c

PROGRAMMING FOR PROBLEM SOLVING

I B. TECH- II SEMESTER

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
A5CS01	ESC	3	-	-	3	30	70	100
Contact Classes: 64	Tutorial Classes: Nil	Practical Classes: Nil				Total Classes: 64		

COURSE OBJECTIVES

1. To impart knowledge about problem solving and algorithmic thinking.
2. To familiarize with the syntax and semantics of C programming language.
3. To learn the usage of structured programming approach in solving problems.
4. To use arrays, pointers, strings and structures in solving problems.
5. To understand how to solve problems related to matrices, Searching and sorting.

COURSE OUTCOMES

At the end of the course, student will be able to:

1. Apply algorithmic thinking to understand, define and solve problems
2. Develop computer programs using programing constructs and control structures
3. Decompose a problem into functions to develop modular reusable code.
4. Use arrays, pointers, strings and structures to formulate algorithms and programs.
5. Use files to perform read and write operations.

UNIT - I	INTRODUCTION - PROBLEM SOLVING AND ALGORITHMIC THINKING	CLASSES: 12

Problem Solving and Algorithmic Thinking Overview – Problem Definition, logical reasoning, Algorithm definition, practical examples, properties, representation, flowchart, algorithms vs programs.
Algorithmic Thinking – Constituents of algorithms - Sequence, Selection and Repetition, input-output; Computation – expressions, logic; Problem Understanding and Analysis – problem definition, variables, name binding, data organization: lists, arrays etc. algorithms to programs.

UNIT - II	OPERATORS, EXPRESSIONS AND CONTROL STRUCTURES	CLASSES: 15

Introduction to C language: Structure of C programs, data types, data inputs, output statements, Operators, precedence and associativity, evaluation of expressions, type conversions in expressions.
Control structures: Decision statements; if and switch statement; Loop control statements: while, for and do while loops, jump statements, break, continue, goto statements.

UNIT - III	ARRAYS AND FUNCTIONS	CLASSES: 17

Arrays: Concepts, One dimensional array, declaration and initialization of one dimensional arrays, two dimensional arrays, initialization and accessing, multi dimensional arrays, Basic Searching Algorithms: Linear and Binary search
Functions: User defined and built-in Functions, storage classes, Parameter passing in functions, call by value, call by reference, Passing arrays to functions, Recursion as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Towers of Hanoi etc.

UNIT - IV	STRINGS AND POINTERS	CLASSES: 10

Strings: Arrays of characters, variable length character strings, inputting character strings, character library functions, string handling functions.
Pointers: Pointer basics, pointer arithmetic, pointers to pointers, generic pointers, array of pointers,

functions returning pointers, Dynamic memory allocation.

UNIT - V	STRUCTURES AND FILE HANDLING	CLASSES: 10

Structures and unions: Structure definition, initialization, accessing structures, nested structures, arrays of structures, structures and functions, self-referential structures, unions, typedef , enumerations.
File handling: command line arguments, File modes, basic file operations read, write and append, example programs

TEXT BOOKS:

1. Riley DD, Hunt K.A. Computational Thinking for the Modern Problem Solver. CRC press, 2014 Mar 27.
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)
3. Byron Gottfried, "Programming with C", Schaum's Outlines Series, McGraw Hill Education, 3rd edition, 2017.

REFERENCE BOOKS:

1. W. Kernighan Brian, Dennis M. Ritchie, "The C Programming Language", PHI Learning, 2nd Edition, 1988.
2. Yashavant Kanetkar, "Exploring C", BPB Publishers, 2nd Edition, 2003.
3. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014.
4. R. S. Bichkar, "Programming with C", Universities Press, 2nd Edition, 2012.
5. Dey Pradeep, Manas Ghosh, "Computer Fundamentals and Programming in C", Oxford University Press, 2nd Edition, 2006.
6. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.

WEB REFERENCES:

1. https://en.wikipedia.org/wiki/Computational_thinking
2. https://nptel.ac.in/courses/106/104/106104128/
3. https://en.cppreference.com/w/c/language
4. https://www.learn-c.org/

E-TEXT BOOKS:

1. https://slidelegend.com/queue/computational-thinking-for-the-modern-problem-solver_59d6f01e1723ddb0c7a0df47.html
2. http://flowgorithm.altervista.org/#elf_l1_Lw
3. http://www.freebookcentre.net/Language/Free-C-Programming-Books-Download.htm

MOOC COURSE

1. https://www.coursera.org/learn/computational-thinking-problem-solving
2. https://onlinecourses.nptel.ac.in/noc18_cs33/preview
3. https://www.alison.com/courses/Introduction-to-Programming-in-c
4. http://www.ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-s096-effective-programming-in-c-and-c-january-iap-2014/index.htm