

PROGRAMMING FOR PROBLEM SOLVING

I Semester: AE / ME II Semester: CSE / IT / ECE / EEE / CE								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSB01	Foundation	L	T	P	C	CIA	SEE	Total
		3	0	0	3	30	70	100
Contact Classes: 45		Tutorial Classes: Nil		Practical Classes: Nil			Total Classes: 45	
<p>OBJECTIVES:</p> <p>The course should enable the students to:</p> <ol style="list-style-type: none"> I. Learn adequate knowledge by problem solving techniques. II. Understand programming skills using the fundamentals and basics of C Language. III. Improve problem solving skills using arrays, strings, and functions. IV. Understand the dynamics of memory by pointers. V. Study files creation process with access permissions. <p>COURSE OUTCOMES (COs):</p> <ol style="list-style-type: none"> 1. Describe the concept of computer system, analyze a given problem, develop an algorithm, fundamental programming constructs, identify data representation formats, describe operators and their precedence, associativity. 2. Understand branching and loop statements. 3. Describe the concept of homogeneous derived data types, strings and functions. 4. Understand pointers and heterogeneous data types. 5. Describe the concept of file system. <p>COURSE LEARNING OUTCOMES (CLOs):</p> <ol style="list-style-type: none"> 1. Identify and understand the working of key components of a computer system. 2. Analyze a given problem and develop an algorithm to solve the problem. 3. Describe the fundamental programming constructs and articulate how they are used to develop a program with a desired runtime execution flow. 4. Gain knowledge to identify appropriate C language constructs to write basic programs. 5. Identify the right data representation formats based on the requirements of the problem. 6. Describe the operators, their precedence and associativity while evaluating expressions in program statements. 7. Understand branching statements, loop statements and use them in problem solving. 8. Learn homogenous derived data types and use them to solve statistical problems. 9. Identify the right string function to write string programs. 10. Understand procedural oriented programming using functions. 11. Understand how recursion works and write programs using recursion to solve problems. 12. Differentiate call by value and call by reference parameter passing mechanisms. 13. Understand storage classes and preprocessor directives for programming. 14. Understand pointers conceptually and apply them in C programs. 15. Distinguish homogenous and heterogeneous data types and apply them in solving data processing applications. 16. Explain the concept of file system for handling data storage and apply it for solving problems. 17. Differentiate text files and binary files and write the simple C programs using file handling functions. 18. Apply the concepts to solve real-time applications using the features of C language. 19. Gain knowledge to identify appropriate searching and sorting techniques by calculating time complexity for problem solving. 20. Possess the knowledge and skills for employability and to succeed in national and international level competitive examinations. 								

MODULE - I	INTRODUCTION	Classes: 10
Introduction to Programming: Computer system, components of a computer system, computing environments, computer languages, creating and running programs, algorithms, flowcharts; Introduction to C language: Computer languages, History of C, basic structure of C programs, process of compiling and running a C program, C tokens, keywords, identifiers, constants, strings, special symbols, variables, data types; Operators and expressions.		
MODULE - II	CONTROL STRUCTURES	Classes: 08
Conditional Control structures: Decision statements; Simple if, if-else, else if ladder, Nested if and Case Statement-switch statement; Loop control statements: while, for and do while loops. jump statements, break, continue, goto statements		
MODULE - III	ARRAYS AND FUNCTIONS	Classes: 10
Arrays: Concepts, one dimensional arrays, declaration and initialization of one dimensional arrays, two dimensional arrays, initialization and accessing, multi-dimensional arrays; Strings: Arrays of characters, variable length character strings, inputting character strings, character library functions, string handling functions. Functions: Need for user defined functions, function declaration, function prototype, category of functions, inter function communication, function calls, parameter passing mechanisms, recursion, passing arrays to functions, passing strings to functions, storage classes, preprocessor directive		
MODULE - IV	STRUCTURES, UNIONS AND POINTERS	Classes: 09
Structures and unions: Structure definition, initialization, accessing structures, nested structures, arrays of structures, structures and functions, passing structures through pointers, self-referential structures, unions, bit fields, typedef, enumerations; Pointers: Pointer basics, pointer arithmetic, pointers to pointers, generic pointers, array of pointers, pointers and arrays, pointers as functions arguments, functions returning pointers. Dynamic memory allocation: Basic concepts, library functions.		
MODULE - V	FILE HANDLING AND BASIC ALGORITHMS	Classes: 08
Files: Streams, basic file operations, file types, file opening modes, input and output operations with files, special functions for working with files, file positioning functions, command line arguments. Searching, basic sorting algorithms (bubble, insertion, selection), algorithm complexity through example programs (no formal definitions required).		
Text Books:		
<ol style="list-style-type: none"> 1. Byron Gottfried, "Programming with C", Schaum's Outlines Series, McGraw Hill Education, 3rd Edition, 2017. 2. E. Balagurusamy, "Programming in ANSI C", McGraw Hill Education, 6th Edition, 2012. 		
Reference Books:		
<ol style="list-style-type: none"> 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://www.bfoit.org/itp/Programming.html>
2. <https://www.khanacademy.org/computing/computer-programming>
3. <https://www.edx.org/course/programming-basics-iitbombayx-cs101-1x-0>
4. <https://www.edx.org/course/introduction-computer-science-harvardx-cs50x>

E-Text Books:

1. <http://www.freebookcentre.net/Language/Free-C-Programming-Books-Download.htm>
2. <http://www.imada.sdu.dk/~svalle/courses/dm14-2005/mirror/c/>
3. <http://www.enggnotebook.weebly.com/uploads/2/2/7/1/22718186/ge6151-notes.pdf>

MOOC Course

1. <https://www.alison.com/courses/Introduction-to-Programming-in-c>
2. <http://www.ocw.mit.edu/courses/electrical-engineering-and-computer-science/6s096-effective-programming-in-c-and-c-january-iap-2014/index.html>