

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

I Semester: Common for AE / ME II Semester: Common for CSE / ECE / EEE / CE / IT								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSB02	Foundation	L	T	P	C	CIA	SEE	Total
		-	-	4	2	30	70	100
Contact Classes: Nil		Tutorial Classes: Nil		Practical Classes: 36			Total Classes: 36	
OBJECTIVES:								
<p>The course should enable the students to:</p> <ol style="list-style-type: none"> I. Formulate problems and implement algorithms using C programming language. II. Develop programs using decision structures, loops and functions. III. Learn memory allocation techniques using pointers. IV. Use structured programming approach for solving of computing problems in real world. 								
COURSE OUTCOMES (COs):								
<ol style="list-style-type: none"> 1. Understand the concept of basics of C, data types and variables.. 2. Understand the concept of operators, precedence of operators, conditional statements and looping statements. 3. Explore the concept of strings, functions, recursive functions and differences between call by value and call by reference. 4. Explore the concept of storage classes, preprocessor directives, pointers and files. 5. Understand the concept of file handling functions, searching and sorting methods and real time applications of C. 								
COURSE LEARNING OUTCOMES (CLOs):								
<ol style="list-style-type: none"> 1. Analyze a given problem and develop an algorithm to solve the problem. 2. Describe the fundamental programming constructs and articulate how they are used to develop a program. 3. Gain knowledge to identify appropriate C language constructs to write basic programs. 4. Identify the right data representation formats based on the requirements of the problem. 5. Describe the operators, their precedence and associativity while evaluating expressions in program statements. 6. Understand branching statements, loop statements and use them in problem solving. 7. Learn homogenous derived data types and use them to solve statistical problems. 8. Identify the right string function to write string programs. 9. Understand procedural oriented programming using functions. 10. Understand how recursion works and write programs using recursion to solve problems. 11. Differentiate call by value and call by reference parameter passing mechanisms. 12. Understand storage classes and preprocessor directives for programming. 13. Understand pointers conceptually and apply them in C programs. 14. Distinguish homogenous and heterogeneous data types and apply them in solving data processing applications. 15. Explain the concept of file system for handling data storage and apply it for solving problems. 16. Differentiate text files and binary files and write the simple C programs using file handling functions. 17. Gain knowledge to identify appropriate searching and sorting techniques by calculating time complexity for problem solving. 18. Apply the concepts to solve real-time applications using the features of C language. 								

LIST OF EXPERIMENTS

Week-1	OPERATORS AND EVALUATION OF EXPRESSIONS
	<p>a. Write a C program to check whether a number is even or odd using ternary operator.</p> <p>b. Write a C program to perform the addition of two numbers without using + operator.</p> <p>c. Write a C program to evaluate the arithmetic expression $((a + b / c * d - e) * (f - g))$. Read the values a, b, c, d, e, f, g from the standard input device.</p> <p>d. Write a C program to find the sum of individual digits of a 3 digit number.</p> <p>e. Write a C program to read the values of x and y and print the results of the following expressions in one line:</p> <p style="padding-left: 20px;">i. $(x + y) / (x - y)$ ii. $(x + y)(x - y)$</p>
Week-2	CONTROL STRUCTURES
	<p>a. Write a C program to find the sum of individual digits of a positive integer.</p> <p>b. A Fibonacci sequence is defined as follows: The first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.</p> <p>c. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.</p> <p>d. A character is entered through keyboard. Write a C program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters.</p> <p>Characters ASCII values</p> <p>A – Z 65 – 90</p> <p>a – z 97 – 122</p> <p>0 – 9 48 – 57</p> <p>Special symbols 0 – 47, 58 – 64, 91 – 96, 123 – 127</p> <p>If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Write a C program to determine how much profit or loss incurred in percentage.</p>
Week-3	CONTROL STRUCTURES
	<p>a. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use switch statement).</p> <p>b. Write a C program to calculate the following sum: sum = $1 - x^2 / 2! + x^4 / 4! - x^6 / 6! + x^8 / 8! - x^{10} / 10!$</p> <p>c. Write a C program to find the roots of a quadratic equation.</p> <p>d. Write a C program to check whether a given 3 digit number is Armstrong number or not.</p> <p>e. Write a C program to print the numbers in triangular form</p> <pre style="padding-left: 20px;"> 1 1 2 1 2 3 1 2 3 4 </pre>
Week-4	ARRAYS
	<p>a. Write a C program to find the second largest integer in a list of integers.</p> <p>b. Write a C program to perform the following:</p> <p style="padding-left: 20px;">i. Addition of two matrices</p> <p style="padding-left: 20px;">ii. Multiplication of two matrices</p> <p>c. Write a C program to count and display positive, negative, odd and even numbers in an array.</p> <p>d. Write a C program to merge two sorted arrays into another array in a sorted order.</p> <p>e. Write a C program to find the frequency of a particular number in a list of integer.</p>

Week-5	STRINGS
<p>a. Write a C program that uses functions to perform the following operations:</p> <ol style="list-style-type: none"> i. To insert a sub string into a given main string from a given position. ii. To delete n characters from a given position in a given string. <p>b. Write a C program to determine if the given string is a palindrome or not.</p> <p>c. Write a C program to find a string within a sentence and replace it with another string.</p> <p>d. Write a C program that reads a line of text and counts all occurrence of a particular word.</p> <p>e. Write a C program that displays the position or index in the string S where the string T begins, or 1 if S doesn't contain T.</p>	
Week-6	FUNCTIONS
<p>a. Write C programs that use both recursive and non-recursive functions</p> <ol style="list-style-type: none"> i. To find the factorial of a given integer. ii. To find the greatest common divisor of two given integers. <p>b. Write C programs that use both recursive and non-recursive functions</p> <ol style="list-style-type: none"> i. To print Fibonacci series. ii. To solve towers of Hanoi problem. <p>c. Write a C program to print the transpose of a given matrix using function.</p> <p>d. Write a C program that uses a function to reverse a given string.</p>	
Week-7	POINTERS
<p>a. Write a C program to concatenate two strings using pointers.</p> <p>b. Write a C program to find the length of string using pointers.</p> <p>c. Write a C program to compare two strings using pointers.</p> <p>d. Write a C program to copy a string from source to destination using pointers.</p> <p>e. Write a C program to reverse a string using pointers.</p>	
Week-8	STRUCTURES AND UNIONS
<p>a. Write a C program that uses functions to perform the following operations:</p> <ol style="list-style-type: none"> i. Reading a complex number ii. Writing a complex number iii. Addition and subtraction of two complex numbers iv. Multiplication of two complex numbers. Note: represent complex number using a structure. <p>b. Write a C program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.</p> <p>c. Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.</p> <p>d. Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.</p> <p>e. Write a C program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.</p>	
Week-9	ADDITIONAL PROGRAMS
<p>a. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+\dots+x^n$. For example: if n is 3 and x is 5, then the program computes $1+5+25+125$. Print x, n, the sum. Perform error checking. For example, the formula does not make sense for negative exponents – if n is less than 0. Have your program print an error message if $n < 0$, then go back and read in the next pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too.</p>	

	<p>b. 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the 2's complement of a binary number.</p> <p>c. Write a C program to convert a Roman numeral to its decimal equivalent. E.g. Roman number CD is equivalent to 400.</p>
Week-10	PREPROCESSOR DIRECTIVES
	<p>a. Define a macro with one parameter to compute the volume of a sphere. Write a C program using this macro to compute the volume for spheres of radius 5, 10 and 15 meters.</p> <p>b. Define a macro that receives an array and the number of elements in the array as arguments. Write a C program for using this macro to print the elements of the array.</p> <p>c. Write symbolic constants for the binary arithmetic operators +, -, *, and /. Write a C program to illustrate the use of these symbolic constants.</p>
Week-11	FILES
	<p>a. Write a C program to display the contents of a file.</p> <p>b. Write a C program to copy the contents of one file to another.</p> <p>c. Write a C program to reverse the first n characters in a file, where n is given by the user.</p> <p>d. Two files DATA1 and DATA2 contain sorted lists of integers. Write a C program to merge the contents of two files into a third file DATA i.e., the contents of the first file followed by those of the second are put in the third file.</p> <p>e. Write a C program to count the no. of characters present in the file.</p>
Week-12	COMMAND LINE ARGUMENTS
	<p>a. Write a C program to read arguments at the command line and display it.</p> <p>b. Write a C program to read two numbers at the command line and perform arithmetic operations on it.</p> <p>c. Write a C program to read a file name at the command line and display its contents.</p>
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. B. A. Forouzan, R. F. Gillberg, "C Programming and Data Structures", Cengage Learning, India, 3rd Edition, 2014. 2. W. Kernighan Brian, Dennis M. Ritchie, "The C Programming Language", PHI Learning, 2nd Edition, 1988. 3. Yashavant Kanetkar, "Exploring C", BPB Publishers, 2nd Edition, 2003. 4. Schildt Herbert, "C: The Complete Reference", Tata McGraw Hill Education, 4th Edition, 2014. 5. R. S. Bichkar, "Programming with C", Universities Press, 2nd Edition, 2012. 6. Dey Pradeep, Manas Ghosh, "Computer Fundamentals and Programming in C", Oxford University Press, 2nd Edition, 2006. 7. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014. 	
Web References:	
<ol style="list-style-type: none"> 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:	
<ol style="list-style-type: none"> 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 	

