INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTION

Course Title	**PROGRAMMING FOR PROBLEM SOLVING LABORATORY**				
Course Code	ACSB02				
Programme	B. Tech				
Semester	ONE				
Course Type	Foundation				
Regulation	IARE - R18				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	-	-	-	4	2
Chief Coordinator	Dr. M Purushotham Reddy, Associate Professor				

I. COURSE OVERVIEW:

The course covers the basics of programming and demonstrates fundamental programming techniques, customs and terms including the most common library functions and the usage of the preprocessor. This course helps the students in gaining the knowledge to write simple C language applications, mathematical and engineering problems. This course helps to undertake future courses that assume this programming language as a background in computer programming. Topics include variables, data types, functions, control structures, pointers, strings, arrays and dynamic allocation principles. This course in reached to student by power point presentations, lecture notes, and lab involve the problem solving in mathematical and engineering areas.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites	Credits
-	-	-	Basic Programming Concepts	-

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Programming for Problem Solving Laboratory	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

	Open Ended Experiments		Demo Video		Lab Worksheets		Viva Questions

V. EVALUATION METHODOLOGY:

Each laboratory will be evaluated for a total of 100 marks consisting of 30 marks for internal assessment and 70 marks for semester end lab examination. Out of 30 marks of internal assessment, continuous lab assessment will be done for 20 marks for the day to day performance and 10 marks for the final internal lab assessment.

Semester End Examination (SEE): The semester end lab examination for 70 marks shall be conducted by two examiners, one of them being Internal Examiner and the other being External Examiner, both nominated by the Principal from the panel of experts recommended by Chairman, BOS.

The emphasis on the experiments is broadly based on the following criteria:

20 %	To test the preparedness for the experiment.
20 %	To test the performance in the laboratory.
20 %	To test the calculations and graphs related to the concern experiment.
20 %	To test the results and the error analysis of the experiment.
20 %	To test the subject knowledge through viva – voce.

Continuous Internal Assessment (CIA):

CIA is conducted for a total of 30 marks (Table 1), with 20 marks for continuous lab assessment during day to day performance, 10 marks for final internal lab assessment.

Table 1: Assessment pattern for CIA

Component	**Laboratory**		**Total Marks**
Type of Assessment	Day to day performance	Final internal lab assessment	
CIA Marks	20	10	30

Continuous Internal Examination (CIE):

One CIE exams shall be conducted at the end of the 16th week of the semester. The CIE exam is conducted for 10 marks of 3 hours duration.

Preparation	**Performance**	**Calculations and Graph**	**Results and Error Analysis**	**Viva**	**Total**
2	2	2	2	2	10

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		**Strength**	**Proficiency assessed by**
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Videos

Program Outcomes		Strength	Proficiency assessed by
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2	Lab Exercises
PO 3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	3	Lab Exercises
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	3	Videos

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Strength	Proficiency assessed by
PSO 1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	2	Assignment
PSO 2	**Problem Solving Skills**: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.	3	Assignment / Lab Exercises
PSO 3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	1	Lab Exercises

3 = High; 2 = Medium; 1 = Low

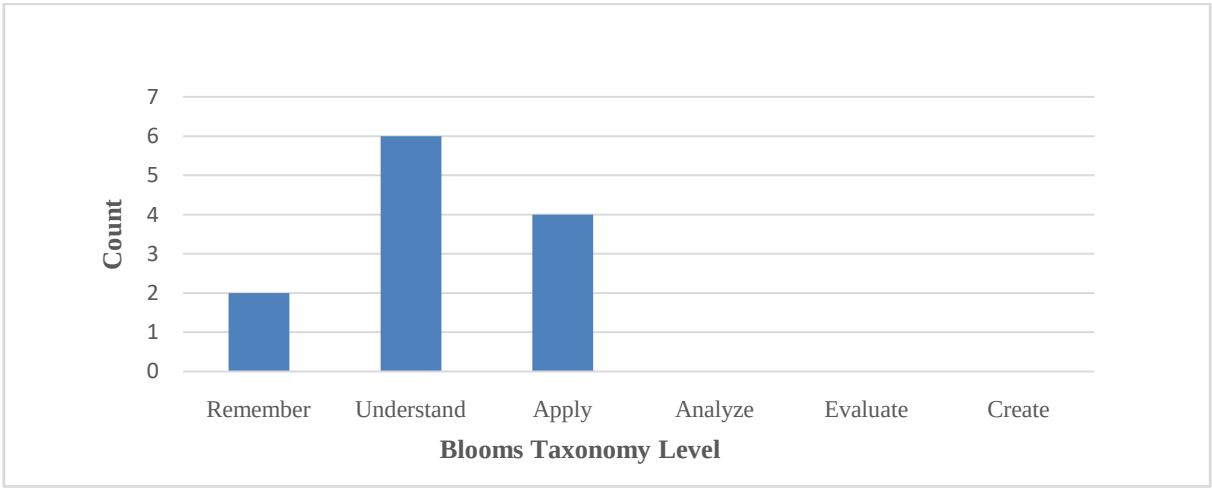
VIII. COURSE OBJECTIVES:

The course should enable the students to:	
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.

IX. COURSE OUTCOMES:

After successful completion of the course, students will be able to:		
CO No	**Course Outcomes**	**Knowledge Level (Bloom's Taxonomy)**
CO 1	**Develop** the algorithms and draw flowcharts for solving Mathematical and Engineering problems.	Apply
CO 2	**Identify, compile and debug** programs in C language.	Apply
CO 3	**Outline**different data types in a computer program.	Understand
CO 4	**Construct** programs involving decision structures and loops.	Apply
CO 5	**Explain** the difference between call by value and call by reference.	Understand
CO 6	**Interpret** the various types of functions, parameters, and return values for complex problem solving.	Understand
CO 7	**Demonstrate** the working of arrays, character strings and array of strings.	Understand
CO 8	**Illustrate** the dynamics of memory by the use of pointers.	Understand
CO 9	**Define** data types and use structures, unions and enumerations to solve problems.	Remember
CO 10	**Interpret** file input and output functions to do integrated programming.	Understand
CO 11	**Utilize** the algorithms in "C" language to real-life computational problems.	Apply
CO 12	**Show** confidence for self-education and ability for life-long learning needed for computer language.	Remember

COURSE KNOWLEDGE COMPETENCY LEVELS



X. JUSTIFICATIONS FOR CO – (PO, PSO) MAPPING

Course Outcomes	POs / PSOs	Justification for mapping (Students will be able to)	No. of key competencies
CO 1	**PO 1**	Developing algorithms and draw flowcharts for solving **mathematical** and **engineering problems** related to **areas of computer science.**	3

	PO 2	Understand the various symbols to **draw** a flowchart, **identify** the appropriate symbols to solve a problem, then **formulate** the solution, and **interpret** the result for the **improvement** of the **solution**.	6
	PSO 1	Understand the features of procedural programming for **designing** and **analyzing** computer programs for**problem-solving.**	3
CO 2	**PO 1**	Apply the knowledge of **mathematics**,**C language fundamentals**to design, develop, and debug programs to solve **engineering** problems.	3
	PO 2	Understand the **problem statement**, identify the **data requirements**, **design**, and **develop** a system for an engineering problem,**validate**and **interpret** the results.	6
	PSO 1	**Understand** automatic type conversion rules to **determine** the magnitude and precision of a mixed datatype expression in the areas of **software development**.	3
CO 3	**PO 1**	Describe the **fundamental programming** constructs, and articulate how they are used to **develop a program** with a desired runtime execution flow.	2
	PO 2	**Identify** the appropriate datatypes to **formulate**, **develop** and **analyze** the solution to achieve **engineering** objectives.	5
	PO 3	Recognize right **data representation** formats based on the **requirements**for **developing programs** in **real-time scenarios**by **managing** the **design process**, and communicating effectively with **engineering** community.	7
	PO 5	Describe the operators, their precedence, and associativity while evaluating expressions in **software program**.	1
CO 4	**PO 1**	Understand branching statements, loop statements, and apply the fundamentals of **mathematics**, **science** and **engineering**.	3
	PO 2	Understand the **problem statement**, **control** the flow of data, **design** the solution and **analyze** the same to **validate** the results in a program to solve **complex** engineering problems.	6
	PO 3	**Recognize** an appropriate control structure to **design** and **develop** a solution for a **real-time** scenario, and **communicating** effectively with engineering community.	5
CO 5	**PO 1**	Make use of**engineering techniques**to design and develop solutions for real-time **computational problems**.	2
CO 6	**PO 1**	Recognize the importance of recursionfor developing programs in real-time scenarios using principles of **mathematics**, and **engineering fundamentals**.	2
	PO 2	Understand the various kinds of **functions**, **identify** the suitable type of function to **solve** a problem, **formulate** the solution, and **interpret** the result for the **improvement** of the solution.	6
	PO 5	Apply techniques of **structured decomposition** to dividea problem into smaller pieces with an understanding of its limitations.	1
CO 7	**PO 1**	Extend the focus on the usage of derived data types as a basic building block in problem solving using principles of **mathematics**, **science**, and **engineering fundamentals**.	3
	PO 2	**Recognize** the multidimensional array **representation**, determine its use to **solve** a problem, **articulate** the solution, and **evaluate** the outcome for solution **enhancement**.	6
	PO 5	Identify the appropriate string function to write **C programs** which need string manipulations.	1

CO 8	**PO 1**	Recognize the importanceof storage classes and preprocessor directives in the solving of problems incorporating **mathematics**, and **engineering principles**.\	2
	PO 5	Understand pointers conceptually and apply them in modeling a **complex engineering** activity.	1
CO 9	**PO 1**	Extend the focus on the usage of heterogeneous data types as a **basic building block** in problem solving using principles of **science**, and **engineering** fundamentals.	3
	PO 2	**Recognize** the representation of the structure, **assess**in solving a problem, **express the solution**, and **analyze** the result for **solution enhancement**.	5
	PO 5	Understand unions conceptually and apply them in modeling a complex **engineering activity**.	1
CO 10	**PO 1**	Make a use of an appropriate type of file to store a large volume of **persistent data** and give solution to **engineering problems**.	2
	PO 5	To identify appropriate mode to access a file and run the same **program** multiple times.	1
CO 11	**PO 5**	Develop, choose and implement suitable methods, tools and modern engineering for **problem solving**.	1
	PSO 1	**Identify** tasks in which the numerical techniques are **applicable**, **develop** programs, and hence use computers effectively to solve **real-time applications**.	4
CO 12	**PO12**	**Realize** the need and the desire to **train** and **invest** in autonomous and **lifelong learning** in the widest sense of **technical transition** to achieve **employability expertise**and excel advanced **engineering concepts**.	7

3 = High; 2 = Medium; 1 = Low

XI. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES

Course Outcomes	Program Outcomes				Program Specific Outcomes		
	PO1	**PO2**	**PO3**	**PO5**	**PSO1**	**PSO2**	**PSO3**
CO 1	3	3	2		2	3	
CO 2	3	3	2		2	3	
CO 3	3	2	2	1	2	3	1
CO 4	3	3	2	2	2	3	
CO 5	2	3	3			3	1

3 = High; 2 = Medium; 1 = Low

XII. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Learning Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO 1	3	2											2	2	

Course Learning Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 2	3	2											3	2	
CO 3		3	2										1	3	
CO 4		3											3		
CO 5	2	3	2											3	
CO 6		3	2											3	
CO 7	3	2	1										2	3	
CO 8	2	3	1										2	3	
CO 9	2	3	1		1								1	3	1
CO 10		2	3										2	3	1
CO 11		2	3										3	2	
CO 12	3	2			2									3	
CO 13	3	2	2										2	3	
CO 14	2	3											3		
CO 15	3	2			2								1	1	
CO 16	3	2											2	3	
CO 17		3	3											3	
CO 18	1		3												1

3 = High; 2 = Medium; 1 = Low

XIII. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 1,PO 2, PO 3,PO 5	SEE Exams	PO 1,PO 2, PO 3,PO 5	Assignments	PO 1,PO 2, PO 3,PO 5	Seminars	-
Laboratory Practices	PO 1,PO 2, PO 3,PO 5	Student Viva	PO 1,PO 2, PO 3,PO 5	Mini Project	-	Certification	-

XIV. ASSESSMENT METHODOLOGIES – INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XV. SYLLABUS

Week-1	OPERATORS AND EVALUATION OF EXPRESSIONS

a. Write a C program to check whether a number is even or odd using ternary operator.
b. Write a C program to perform the addition of two numbers without using + operator.
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.
d. Write a C program to find the sum of individual digits of a 3 digit number.
e. Write a C program to read the values of x and y and print the results of the following expressions in one line:
   i. (x + y) / (x - y)
   ii. (x + y)(x - y)

Week-2	CONTROL STRUCTURES

a. Write a C program to find the sum of individual digits of a positive integer.
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.
c. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
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.
   Characters ASCII values
   A – Z 65 – 90 a – z 97 – 122 0 – 9 48 – 57
   Special symbols 0 – 47, 58 – 64, 91 – 96, 123 – 127
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.

Week-3	CONTROL STRUCTURES

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).
b. Write a C program to calculate the following sum:
   sum = 1 – x2 /2! + x4 /4! – x6 /6! +x8 /8! – x10/10!
c. Write a C program to find the roots of a quadratic equation.
d. Write a C program to check whether a given 3 digit number is Armstrong number or not.
e. Write a C program to print the numbers in triangular form
   1
   1 2
   1 2 3

Week-4	ARRAYS

a. Write a C program to find the second largest integer in a list of integers.
b. Write a C program to perform the following:
   i. Addition of two matrices
   ii. Multiplication of two matrices
c. Write a C program to count and display positive, negative, odd and even numbers in an array.
d. Write a C program to merge two sorted arrays into another array in a sorted order.
e. Write a C program to find the frequency of a particular number in a list of integer.

Week-5	STRINGS

a. Write a C program that uses functions to perform the following operations:
   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.
b. Write a C program to determine if the given string is a palindrome or not.
c. Write a C program to find a string within a sentence and replace it with another string.
d. Write a C program that reads a line of text and counts all occurrence of a particular word.
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.

Week-6	FUNCTIONS

a. Write C programs that use both recursive and non-recursive functions
   i. To find the factorial of a given integer.
   ii. To find the greatest common divisor of two given integers.
b. Write C programs that use both recursive and non-recursive functions
   i. To print Fibonacci series.
   ii. To solve towers of Hanoi problem.
c. Write a C program to print the transpose of a given matrix using function.
d. Write a C program that uses a function to reverse a given string.

Week-7	POINTERS

a. Write a C program to concatenate two strings using pointers.
b. Write a C program to find the length of string using pointers.
c. Write a C program to compare two strings using pointers.
d. Write a C program to copy a string from source to destination using pointers.
e. Write a C program to reverse a string using pointers.

Week-8	STRUCTURES AND UNIONS

a. Write a C program that uses functions to perform the following operations:
   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.
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.
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.
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.
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.

Week-9	ADDITIONAL PROGRAMS

a. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+\ldots\ldots\ldots\ldots+x^n$. For example: if n is 3 and x is 5, then the program computes1+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.
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 the2"s complement of a binary number.

c. Write a C program to convert a Roman numeral to its decimal equivalent. E.g. Roman number CD is equivalent to 400.

Week-10	**PREPROCESSOR DIRECTIVES**

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.
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.
c. Write symbolic constants for the binary arithmetic operators +, -, *, and /. Write a C program to illustrate the use of these symbolic constants.

Week-11	**FILES**

a. Write a C program to display the contents of a file.
b. Write a C program to copy the contents of one file to another.
c. Write a C program to reverse the first n characters in a file, where n is given by the user.
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.
e. Write a C program to count the no. of characters present in the file.

Week-12	**COMMAND LINE ARGUMENTS**

a. Write a C program to read arguments at the command line and display it.
b. Write a C program to read two numbers at the command line and perform arithmetic operations on it.
c. Write a C program to read a file name at the command line and display its contents.

Text Books:

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:

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:

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

XVI. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Week No.	Topics to be covered	Course Outcomes	Reference
1	Operators and Evaluation of Expressions	CO 1, CO 2, CO 3, CO 4, CO 5	T1:2.11-3.5
2	Control Structures	CO 1, CO 2, CO 3, CO 6	T1: 6.1-6.11
3	Control Structures	CO 1, CO 2, CO 3, CO 6	T1: 6.1-6.11
4	Arrays	CO 1, CO 2, CO 3, CO 7	T1: 9.1-9.4
5	Strings	CO 1, CO 2, CO 3, CO 8	T1: 9.5
6	Functions	CO 1, CO 2, CO 9, CO 10	T1: 7.1-7.6
7	Pointers	CO 1,CO 11, CO 13	T1:10.1-10.10
8	Structures and Unions	CO 14	T1:11.1-11.7
9	Additional Programs	CO 17, CO18	R1:4.2
10	Preprocessor Directives	CO 12	T1: 14.5
11	Files	CO 15, CO 16	T1:12.1-12.6
12	Command Line Arguments	CO 12	T1: 14.2

Prepared by:
Dr. M Purushotham Reddy, Assistant Professor **HOD, CSE**