INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTOR

Course Title	**PROGRAMMING FOR PROBLEM SOLVING**				
Course Code	ACSB01				
Programme	B.Tech				
Semester	I	AE | ME			
	II	CSE | IT | ECE |EEE			
Course Type	Foundation				
Regulation	IARE - R18				
Course Structure	Theory			Practical	
	Lectures	Tutorials	Credits	Laboratory	Credits
	3	0	3	4	2
Course Coordinator	Mr.P Ravinder , Assistant Professor				
Course Faculty	Dr J Sirisha Devi, Associate Professor, CSE Dept Dr R ObulaKonda Reddy Associate Professor , CSE Dept Mrs K Laxmi Narayanamma, Assistant Professor, IT Dept. Mrs B Padmaja Assistant Professor, CSE Dept Dr.M Purushotham Reddy, IT Dept Mr. Ch Suresh Kumar Raju Assistant Professor, CSE Dept.				

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	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✗	Chalk & Talk	✔	Quiz	✔	Assignments	✔	MOOCs
✔	LCD / PPT	✔	Seminars	✗	Mini Project	✔	Videos
✔	Open Ended Experiments						

V. EVALUATION METHODOLOGY:

The course will be evaluated for a total of 100 marks, with 30 marks for Continuous Internal Assessment (CIA) and 70 marks for Semester End Examination (SEE). Out of 30 marks allotted for CIA during the semester, marks are awarded by taking average of two CIA examinations or the marks scored in the make-up examination.

Semester End Examination (SEE): The SEE is conducted for 70 marks of 3 hours duration. The syllabus for the theory courses is divided into five modules and each modules carries equal weightage in terms of marks distribution. The question paper pattern is as follows. Two full questions with "either" or "choice" will be drawn from each module. Each question carries 14 marks. There could be a maximum of two sub divisions in a question.

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

50 %	To test the objectiveness of the concept.
50 %	To test the analytical skill of the concept OR to test the application skill of the concept.

Continuous Internal Assessment (CIA):
CIA is conducted for a total of 30 marks (Table 1), with 20 marks for Continuous Internal Examination (CIE), 05 marks for Quiz and 05 marks for Alternative Assessment Tool (AAT).

Table 1: Assessment pattern for CIA

Component	**Theory**			**Total Marks**
Type of Assessment	CIE Exam	Quiz	AAT	
CIA Marks	20	05	05	30

Continuous Internal Examination (CIE):
Two CIE exams shall be conducted at the end of the 8th and 16th week of the semester respectively. The CIE exam is conducted for 20 marks of 2 hours duration consisting of five descriptive type questions out of which four questions have to be answered where, each question carries 5 marks. Marks are awarded by taking average of marks scored in two CIE exams.

Quiz - Online Examination
Two Quiz exams shall be online examination consisting of 25 multiple choice questions and are to be answered by choosing the correct answer from a given set of choices (commonly four). Such a question paper shall be useful in testing of knowledge, skills, application, analysis, evaluation and understanding of the students. Marks shall be awarded considering the average of two quiz examinations for every course.

Alternative Assessment Tool (AAT)
This AAT enables faculty to design own assessment patterns during the CIA. The AAT converts the classroom into an effective learning centre. The AAT may include tutorial hours/classes, seminars, assignments, term paper, open ended experiments, METE (Modeling and Experimental Tools in Engineering), five minutes video, MOOCs etc. The AAT chosen for this course is given in section XI.

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		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	Assignments/Quiz
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	Assignments/Quiz
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	Assignments
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	Assignments
PO 12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	2	Assignments

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO 1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.	3	Assignments/Quiz
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	Assignments/Quiz
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.	3	Assignments

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES :

The course should enable the students to:	
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.

IX. COURSE OUTCOMES (COs):

Cos	Course Outcome	CLOs	Course Learning Outcome
CO 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.	CLO 1	Identify and understand the working of key components of a computer system.
		CLO 2	Analyze a given problem and develop an algorithm to solve the problem.
		CLO 3	Describe the fundamental programming constructs and articulate how they are used to develop a program with a desired runtime execution flow.
		CLO 4	Gain knowledge to identify appropriate C language constructs to write basic programs.
		CLO 5	Identify the right data representation formats based on the requirements of the problem.
		CLO 6	Describe the operators, their precedence and associativity while evaluating expressions in program statements.
CO 2	Understand branching and loop statements.	CLO 7	Understand branching statements, loop statements and use them in problem solving.
CO 3	Describe the concept of homogeneous derives data types, strings and functions.	CLO 8	Learn homogenous derived data types and use them to solve statistical problems.
		CLO 9	Identify the right string function to write string programs.
		CLO 10	Understand procedural oriented programming using functions.
		CLO 11	Understand how recursion works and write programs using recursion to solve problems.
		CLO 12	Differentiate call by value and call by reference parameter passing mechanisms.
		CLO 13	Understand storage classes and preprocessor directives for programming.
CO 4	Understand pointers and heterogeneous data types.	CLO 14	Understand pointers conceptually and apply them in C programs.
		CLO 15	Distinguish homogenous and heterogeneous data types and apply them in solving data processing applications.
CO 5	Describe the concept of file system.	CLO 16	Explain the concept of file system for handling data storage and apply it for solving problems.
		CLO 17	Differentiate text files and binary files and write the simple C programs using file handling functions.
		CLO 18	Apply the concepts to solve real-time applications using the features of C language.
		CLO 19	Gain knowledge to identify appropriate searching and sorting techniques by calculating time complexity for problem solving.
		CLO 20	Possess the knowledge and skills for employability and to succeed in national and international level competitive examinations.

X. COURSE LEARNING OUTCOMES (CLOs):

CLO Code	CLO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
ACSB01.01	CLO 1	Identify and understand the working of key components of a computer system.	PO 1,PO2	2
ACSB01.02	CLO 2	Analyze a given problem and develop an algorithm to solve the problem.	PO 1,PO2	2
ACSB01.03	CLO 3	Describe the fundamental programming constructs and articulate how they are used to develop a program with a desired runtime execution flow.	PO 1,PO2	2
ACSB01.04	CLO 4	Gain knowledge to identify appropriate C language constructs to write basic programs.	PO 2,PO3	3

CLO Code	CLO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
ACSB01.05	CLO 5	Identify the right data representation formats based on the requirements of the problem.	PO 2,PO12	2
ACSB01.06	CLO 6	Describe the operators, their precedence and associativity while evaluating expressions in program statements..	PO 1,PO2,PO3	3
ACSB01.07	CLO 7	Understand branching statements, loop statements and use them in problem solving.	PO 2	2
ACSB01.08	CLO 8	Learn homogenous derived data types and use them to solve statistical problems.	PO1,PO2,PO3	2
ACSB01.09	CLO 9	Identify the right string function to write string programs.	PO 1,PO2,PO3	2
ACSB01.10	CLO 10	Understand procedural oriented programming using functions.	PO1,PO2,PO3, PO5	2
ACSB01.11	CLO 11	Understand how recursion works and write programs using recursion to solve problems.	PO2 ,PO3	3
ACSB01.12	CLO 12	Differentiate call by value and call by reference parameter passing mechanisms.	PO 2,PO3	2
ACSB01.13	CLO 13	Understand storage classes and preprocessor directives for programming	PO 1,PO2,PO5	3
ACSB01.14	CLO 14	Understand pointers conceptually and apply them in C programs.	PO 1,PO2,PO3	3
ACSB01.15	CLO 15	Distinguish homogenous and heterogeneous data types and apply them in solving data processing applications.	PO 1,PO2	2
ACSB01.16	CLO 16	Explain the concept of file system for handling data storage and apply it for solving problems	PO 1,PO2,PO5	3
ACSB01.17	CLO 17	Differentiate text files and binary files and write the simple C programs using file handling functions.	PO 1,PO2	2
ACSB01.18	CLO 18	Apply the concepts to solve real-time applications using the features of C language.	PO 2, PO 12	2
ACSB01.19	CLO 19	Gain knowledge to identify appropriate searching and sorting techniques by calculating time complexity for problem solving.	PO 2,PO3,PO12	3
ACSB01.20	CLO 20	Possess the knowledge and skills for employability and to succeed in national and international level competitive examinations.	PO 5,PO12	2

3 = High; 2 = Medium; 1 = Low

XI. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES:

Course Outcomes (COs)	Program Outcomes (POs)					Program Specific Outcomes (PSOs)		
	PO 1	PO 2	PO 3	PO 5	PO 12	PSO 1	PSO 2	PSO 3
CO 1	3	3	3		2	3	3	2
CO 2		2					3	
CO 3	3	3	3	3		2	3	1
CO 4	3	3	3			3	3	
CO 5	3	3	3	3	3	2	3	3

3 = High; 2 = Medium; 1 = Low

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

Course Learning Outcomes (CLOs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 1	2	2											2	2	2
CLO 2	2	2											2	2	
CLO 3	2	2											3	2	
CLO 4		3	3										1	3	
CLO 5		2										2	3		
CLO 6	3	3	3											3	
CLO 7		2												3	
CLO 8	2	2	2										2	3	
CLO 9	2	2	2										2	3	
CLO 10	2	2	2		2								1	3	1
CLO 11		2	3										2	3	1
CLO 12		2	2										3	2	
CLO 13	3	3			3									3	
CLO 14	3	3	3										2	3	
CLO 15	2	2											3		
CLO 16	3	3			3								1	1	
CLO 17	2	2											2	3	
CLO 18		2										2			3
CLO 19		3	3									3		3	
CLO 20					2							2			3

3 = High; 2 = Medium; 1 = Low

XIII. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO1,PO2,PO3,PO5,PO12,PSO1,PSO2,PSO3	SEE Exams	PO1,PO2,PO3,PO5,PO12,PSO1,PSO2,PSO3	Assignments	PO1,PO2,PO3,PO5,PO12,PSO1,PSO2,PSO3	Seminars	PO1,PO2,PO3,PO5,PO12,PSO1,PSO2,PSO3
Laboratory Practices	PO1,PO2,PO3,PO5,PO12,PSO1,PSO2,PSO3	Student Viva	PO1,PO2,PO3,PO5,PO12,PSO1,PSO2,PSO3	Mini Project	-	Certification	-
Term Paper	-						

XIV. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE feed Back
✗	Assessment of Mini Projects by Experts		

XV. SYLLABUS

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 BASICALGORITHMS**	**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:		
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. 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.		
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/6-s096-effective-programming-in-c- and-c-january-iap-2014/index.htm

XVI. COURSE PLAN:

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

Lecture No	Topic/s to be covered	Course Learning Outcomes (CLOs)	Reference
1 – 2	Introduction to Computers: computer systems, computing environments, Computer languages, creating and running programs	CLO 1	T2:1.1-1.2
3 – 4	Algorithms, flowcharts; Introduction to C language: Computer languages, History of C, basic structure of C programs, process of compiling and running a C program	CLO 2	T2:2.1-2.2
5 – 6	C tokens, keywords, identifiers, constants, strings	CLO 2	T2:1.4-1.5
7 – 8	Special symbols, variables, data types	CLO 3	T2:2.1-2.2
9 – 10	Operators and expressions	CLO 3	T2: 2.3-2.6,7
11 – 12	Simple if, if-else, else if ladder, Nested if and Case Statement-switch statement	CLO 3	T2:3.1-3.5
13 – 14	While, for and do while loops	CLO 5	T2: 5.2-5.3
15 – 16	Jump statements, break, continue, goto statements	CLO 7	T2: 6.1-6.6
17 – 18	Concepts, one dimensional arrays, declaration and initialization of one dimensional arrays	CLO 9	T2: 6.7
19 – 20	Two dimensional arrays, initialization and accessing	CLO 13	T2: 8.1- 8.3
21-22	Multi-dimensional arrays; Strings: Arrays of characters	CLO 13	T2: 11.1-11.5
23-- 24	Variable length character strings, inputting character strings, character library functions, string handling functions	CLO 15	T2: 4.1-4.5
25	Need for user defined functions, function declaration, function prototype	CLO 15	T1:7 T2: 6.9
26	Category of functions, inter function communication, function calls	CLO 11	T1:10T2:10.1- 10.2
27	Parameter passing mechanisms, recursion, passing arrays to functions, passing strings to functions,	CLO 16	T2:10.3- 10.5
28	Storage classes ,preprocessor directives	CLO 16	T1:8.9
29	Structure definition, initialization, accessing structures, nested structures	CLO 16	T2: 12.3- 12.4
30	Unions, C programming examples, BitFields, typedef, enumerations	CLO 16	T2:12.4
31 – 32	Arrays of structures, structures and functions, passing structures through pointers, self-referential structures	CLO 17	T2:2.1-2.2

33– 34	Unions, bit fields, typedef, enumerations	CLO 17	T2: 2.3- 2.6,7
35 – 36	Pointer basics, pointer arithmetic, pointers to pointers, generic pointers, array of pointers, pointers and arrays	CLO 19	T2:3.1-3.5
37	Pointers as functions arguments, functions returning pointers	CLO 19	T2: 5.2-5.3
38	Dynamic memory allocation: Basic concepts, library functions	CLO 20	T2: 6.1-6.6
39	Streams, basic file operations, file types, file opening modes, input and output operations with files	CLO 20	T2:10.4
40-41	Special functions for working with files, file positioning functions	CLO 21	R3:12.1- 12.3
42	Command line arguments. Searching	CLO 22	R3:12.4
43	Sorting algorithms bubble, insertion, selection	CLO 23	T2:11.4 R7:13.1
44-45	Algorithm complexity through example programs	CLO 23	T2:11.4 R7:13.1

XVII. GAPS IN THE SYLLABUS - TO MEET INDUSTRY / PROFESSION REQUIREMENTS:

S No	Description	Proposed Actions	Relevance With POs	Relevance With PSOs
1	Assist student to design system calls in operating systems	Seminars	PO 1,PO 2	PSO 1
2	Stimulate students to develop graphics programming	Seminars/ NPTEL	PO 2	PSO 1
3	Encourage students to solve real time applications and prepare towards competitive examinations.	1. Build IT 2. Proficiency Test 3. Coding Hackthon/ Competitions	PO 2	PSO 1

XVIII. DESIGN BASED PROBLEMS (DP) / OPEN ENDEDPROBLEM:

1. Develop simple character-based Chess-game supporting standard partial chess moves. Chess board should be 8x8 Cell Board having each Cell of 4 characters. Basic chess board with empty shell should have A…Cell and B... For Black Cell. Wherever any player‟s Game elements such as Rook or Camel or King or Queen is on board Cell then it. Then it should be displayed such as BQN2 or WQN1 which indicated such as Queen of player-2 on black cell or queen of player-1 on white cell. Or Student can use his own conventions. Student should be able to demonstrate 5 moves for each player minimum.
2. (Airline Reservations System) A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system. You are to write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats).
3. Your program should display the following menu of alternatives: Please type 1 for "first class"

   Please type 2 for "economy"

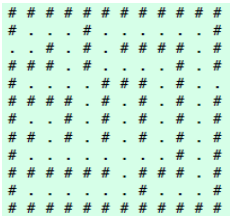
   If the person types 1, then your program should assign a seat in the first class section (seats 1- 5). If the person types 2, then your program should assign a seat in the economy section (seats 6-10). Your program should then print a boarding pass indicating the person's seat number and whether it is in the first class or economy section of the plane.

   Use a single-subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding elements of the array to 1 to indicate that the seat is no longer

   available.

   Your program should, of course, never assign a seat that has already been assigned. When the first

   class section is full, your program should ask the person if it is acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message "Next flight leaves in 3 hours.



(*Maze Traversal*) The following grid is a double-subscripted array representation of a maze.

The # symbols represent the walls of the maze, and the periods (.) represent squares in the possible paths through the maze. There is a simple algorithm for walking through a maze that guarantees finding the exit (assuming there is an exit). If there is not an exit, you will arrive at the starting location again. Place your right hand on the wall to your right and begin walking forward. Never remove your hand from the wall. If the maze turns to the right, you follow the wall to the right. As long as you do not remove your hand from the wall, eventually you will arrive at the exit of the maze. There may be a shorter path than the one you have taken, but you are guaranteed to get out of the maze. Write recursive function maze Traverse to walk through the maze. The function should receive as arguments a 12-by-12 character array representing the maze and the starting location of the maze. As maze Traverse attempts to locate the exit from the maze, it should place the character X in each square in the path. The function should display the maze after each move so the user can watch as the maze is solved.

Prepared by: **HOD, CSE**

Mr. P Ravinder, Assistant Professor