# INSTITUTE OF AERONAUTICAL ENGINEERING

**(Autonomous)**
Dundigal, Hyderabad -500 043

## INFORMATION TECHNOLOGY

## COURSE DESCRIPTOR

| | | | | | | |
|---|---|---|---|---|---|---|
| **Course Title** | **COMPUTER PROGRAMMING LABORATORY** | | | | | |
| **Course Code** | ACS101 | | | | | |
| **Programme** | B.Tech | | | | | |
| **Semester** | I | CSE\| IT \| ECE \| EEE | | | | |
| | II | AE\| CE \| ME | | | | |
| **Course Type** | Foundation | | | | | |
| **Regulation** | IARE - R16 | | | | | |
| **Course Structure** | **Theory** | | | **Practical** | | |
| | **Lectures** | **Tutorials** | **Credits** | **Laboratory** | **Credits** | |
| | - | - | - | 3 | 2 | |
| **Chief Coordinator** | Mr. Ch Suresh Kumar Raju, Assistant Professor | | | | | |
| **Course Faculty** | Dr. B Venkateswara Rao, 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:

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

## 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 16$^{th}$ 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 (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 | Videos |
| 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 (PSOs) | | 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 | **Software Engineering Practices:** The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality service 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 (COs):

| **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 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 |
|---|---|---|---|---|
| ACSB02.01 | CLO 1 | Analyze a given problem and develop an algorithm to solve the problem | PO 1, PO 2 | 2 |
| ACSB02.02 | CLO 2 | Describe the fundamental programming constructs and articulate how they are used to develop a program. | PO 1, PO 2 | 2 |
| ACSB02.03 | CLO 3 | Gain knowledge to identify appropriate C language constructs to write basic programs. | PO 2, PO 3 | 3 |
| ACSB02.04 | CLO 4 | Identify the right data representation formats based on the requirements of the problem. | PO 2 | 2 |
| ACSB02.05 | CLO 5 | Describe the operators, their precedence and associativity while evaluating expressions in program statements.. | PO 1,PO 2, PO 3 | 3 |
| ACSB02.06 | CLO 6 | Understand branching statements, loop statements and use them in problem solving. | PO 2, PO 3 | 2 |
| ACSB02.07 | CLO 7 | Learn homogenous derived data types and use them to solve statistical problems. | PO 1, PO 2, PO 3 | 3 |
| ACSB02.08 | CLO 8 | Identify the right string function to write string programs. | PO 1, PO 2, PO 3 | 3 |
| ACSB02.09 | CLO 9 | Understand procedural oriented programming using functions. | PO 1, PO 2, PO 3,PO5 | 3 |
| ACSB02.10 | CLO 10 | Understand how recursion works and write programs using recursion to solve problems. | PO 2, PO 3 | 2 |
| ACSB02.11 | CLO 11 | Differentiate call by value and call by reference parameter passing mechanisms. | PO 2, PO 3 | 2 |
| ACSB02.12 | CLO 12 | Understand storage classes and preprocessor directives for programming | PO 1, PO 2,PO5 | 3 |
| ACSB02.13 | CLO 13 | Understand pointers conceptually and apply them in C programs. | PO 1, PO 2,PO3 | 3 |
| ACSB02.14 | CLO 14 | Distinguish homogenous and heterogeneous data types and apply them in solving data processing applications. | PO 1, PO 2 | 2 |
| ACSB02.15 | CLO 15 | Explain the concept of file system for handling data storage and apply it for solving problems. | PO 1, PO 2,PO5 | 3 |
| ACSB02.16 | CLO 16 | Differentiate text files and binary files and write the simple C programs using file handling functions. | PO 1, PO 2 | 2 |
| ACSB02.17 | CLO 17 | Gain knowledge to identify appropriate searching and sorting techniques by calculating time complexity for problem solving. | PO 2, PO 3 | 3 |
| ACSB02.18 | CLO 18 | Apply the concepts to solve real-time applications using the features of C language. | PO 1, PO 3 | 2 |

**3 = High; 2 = Medium; 1 = Low**

## X. 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 | 3 | 2 | | | | | | | | | | | 2 | 2 | |
| CLO 2 | 3 | 2 | | | | | | | | | | | 3 | 2 | |
| CLO 3 | | 3 | 2 | | | | | | | | | | 1 | 3 | |
| CLO 4 | | 3 | | | | | | | | | | | 3 | | |
| CLO 5 | 2 | 3 | 2 | | | | | | | | | | | 3 | |
| CLO 6 | | 3 | 2 | | | | | | | | | | | 3 | |
| CLO 7 | 3 | 2 | 1 | | | | | | | | | | 2 | 3 | |
| CLO 8 | 2 | 3 | 1 | | | | | | | | | | 2 | 3 | |
| CLO 9 | 2 | 3 | 1 | | 1 | | | | | | | | 1 | 3 | 1 |
| CLO 10 | | 2 | 3 | | | | | | | | | | 2 | 3 | 1 |
| CLO 11 | | 2 | 3 | | | | | | | | | | 3 | 2 | |
| CLO 12 | 3 | 2 | | | 2 | | | | | | | | | 3 | |
| CLO 13 | 3 | 2 | 2 | | | | | | | | | | 2 | 3 | |
| CLO 14 | 2 | 3 | | | | | | | | | | | 3 | | |
| CLO 15 | 3 | 2 | | | 2 | | | | | | | | 1 | 1 | |
| CLO 16 | 3 | 2 | | | | | | | | | | | 2 | 3 | |
| CLO 17 | | 3 | 3 | | | | | | | | | | | 3 | |
| CLO 18 | 1 | | 3 | | | | | | | | | | | | 1 |

**3 = High; 2 = Medium; 1 = Low**

## XI. ASSESSMENT METHODOLOGIES – DIRECT

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CIE Exams | PO 1, PO 2, PO 3, PO 5, PO 12 | SEE Exams | PO 1, PO 2, PO 3, PO 5, PO 12 | Assignments | PO 1, PO 2, PO 3, PO 5, PO 12 | Seminars | - |
| Laboratory Practices | PO 1, PO 2, PO 3, PO 5, PO 12 | Student Viva | PO 1, PO 2, PO 3, PO 5, PO 12 | Mini Project | - | Certificatio | - |

## XII. ASSESSMENT METHODOLOGIES – INDIRECT

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

## XIII. SYLLABUS

| LIST OF EXPERIMENTS | |
|---|---|
| **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
   1 2 3 4

| | |
|---|---|
| **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<br>c. Write a C program to count and display positive, negative, odd and even numbers in an array.<br>d. Write a C program to merge two sorted arrays into another array in a sorted order.<br>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:<br>i. To insert a sub string into a given main string from a given position.<br>ii. To delete n characters from a given position in a given string.<br>b. Write a C program to determine if the given string is a palindrome or not.<br>c. Write a C program to find a string within a sentence and replace it with another string.<br>d. Write a C program that reads a line of text and counts all occurrence of a particular word.<br>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<br>i. To find the factorial of a given integer.<br>ii. To find the greatest common divisor of two given integers.<br>b. Write C programs that use both recursive and non-recursive functions<br>i. To print Fibonacci series.<br>ii. To solve towers of Hanoi problem.<br>c. Write a C program to print the transpose of a given matrix using function.<br>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.<br>b. Write a C program to find the length of string using pointers.<br>c. Write a C program to compare two strings using pointers.<br>d. Write a C program to copy a string from source to destination using pointers.<br>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:<br>i. Reading a complex number<br>ii. Writing a complex number<br>iii. Addition and subtraction of two complex numbers<br>iv. Multiplication of two complex numbers. Note: represent complex number using a structure.<br>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.<br>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.<br>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.<br>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 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.

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<br>2. http://www.imada.sdu.dk/~svalle/courses/dm14-2005/mirror/c/<br>3. http://www.enggnotebook.weebly.com/uploads/2/2/7/1/22718186/ge6151-notes.pdf |

## XIV. COURSE PLAN:

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

| Week No. | Topics to be covered | Course Learning Outcomes (CLOs) | Reference |
|---|---|---|---|
| 1 | Operators and Evaluation of Expressions | CLO 1, CLO 2, CLO 3, CLO 4 | T2:1.4-1.5 |
| 2 | Control Structures | CLO 5, CLO 6 | T2:3.1-3.5 |
| 3 | Control Structures | CLO 5, CLO 6 | T2: 5.2-5.3 |
| 4 | Arrays | CLO 5, CLO 6, CLO 7 | T2: 6.7 |
| 5 | Strings | CLO 5, CLO 6, CLO 7, CLO 8 | T2: 4.1-4.5 |
| 6 | Functions | CLO 5, CLO 6, CLO 9 | T1:7, 10 |
| 7 | Pointers | CLO 5, CLO 6, CLO 7, CLO 13 | T2:10.3-10.5 |
| 8 | Structures and Unions | CLO 5, CLO 6, CLO 7, CLO 13, | T2: 12.1- 12.4 |
| 9 | Additional Programs | CLO 5, CLO 6, CLO 7 | T2: 6.1-6.6 |
| 10 | Preprocessor Directives | CLO 6, CLO 7, CLO 12 | T1:8 |
| 11 | Files | CLO 6, CLO 7, CLO 15 | T2:10.4 |
| 12 | Command Line Arguments | CLO 6, CLO 7, CLO 15, CLO 16, CLO 17, CLO 18 | R3:12.4 |

## XV. 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 | 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<br>2. Proficiency Test<br>3. Coding Hackthon/ Competitions | PO 2 | PSO 1 |

Mr.Ch Suresh Kumar Raju, Assistant Professor, CSE
Dr. B Venkateswara Rao, Professor

**Prepared by:**

**HOD, FRESHMAN ENGINEERING**