Savitribai Phule Pune University
First Year of Engineering (2024 Pattern)
Course Code: ESC-105-COM Course Name: Fundamentals of Programming Languages

Teaching Scheme	Credit	Examination Scheme
Theory : 2 Hours/Week **Practical : 2 Hours/Week**	**02** **01**	**CCE : 30 Marks** **End – Semester : 70 Marks** **Term Work : 25 Marks**

Prerequisite Courses, if any:

- Basics of Computers
- Basic Mathematics

Companion Course, if any: Fundamentals of Programming Languages Lab

Course Objectives:

1. To understand the fundamental Concepts of C Programming
2. To acquire knowledge and Compare usage of Operators and Expressions in C Programming
3. To apply Control Flow structures in C Programming for Problem solving
4. To design a solution using Arrays, Character and String Arrays in C programming
5. To design a develop solution for simple computational problems using User Defined Functions and structures in C Programming

Course Outcomes:

On completion of the course, students will be able to:

CO1: To **Design** algorithms for simple computational problems.

CO2: To **Use** mathematical, Logical Operators and Expressions.

CO3: To **apply** Control Flow structures for decision making.

CO4: To **design** a solution using Arrays, Character and String Arrays.

CO5: To **D**esign and apply user defined functions and structures.

Unit I	Introduction to Program Planning & C Programming	(06 Hours)

Program Design Tools: Art of Programming through Algorithms, Flowcharts.

Overview of C: History and importance C, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, Data types, Declaration of variables, Storage Class, Assigning Values to variables, Defining Symbolic Constants, declaring a Variable as Constant, Declaring a Variable as Volatile.

#Exemplar/Case Studies	Study of "C" Program compilation Process, testing and debugging.

Unit II	Operators and Expressions	(06 Hours)

Operators and Expressions: Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Bitwise Operators,

Special Operators. Arithmetic Expressions, Evaluation of Expressions, Precedence of Arithmetic Operators, Operator Precedence and Associativity, Mathematical Functions.		
#Exemplar/Case Studies	Study of Infix, Prefix and Postfix expressions.	
Unit III	**Control Flow**	**(06 Hours)**
Decision Making and Branching: Simple If Statement, If-Else,Else-If,Switch Statement, Goto Statement **Decision Making and Looping:** While Statement, Do-While, For Statement, Break and Continue		
#Exemplar/Case Studies	Design simple calculator and Generating a Calendar	
Unit IV	**Arrays**	**(06 Hours)**
Arrays: One Dimensional Arrays, Declaration of One-dimensional Arrays, Initialization of One-dimensional Arrays, Two –dimensional Arrays, Initialization of Two- dimensional Arrays. **Character Arrays and Strings:** Declaration and Initialization String Variables, Reading Strings from Terminal, Writing Strings to Screen, Putting Strings Together, Comparison of Two Strings, Introduction to String handling Functions		
#Exemplar/Case Studies	Matrix multiplication	
Unit V	**User Defined Functions**	**(06 Hours)**
User Defined Functions: Need for User-defined Functions, A Multi-Function Program, Elements of User defined Functions, Definition of Functions, Return Values and their Types, Function Calls, Function Declaration, Category of Functions: No Arguments and no Return Values, Arguments but No Return Values, Arguments with Return values, No Arguments but Returns a Value, Functions that Return Multiple Values, Nesting of Functions, Recursion **Structures :** What is a Structure? Structure Type Declarations, Structure Declarations, Referencing Structure Members, Referencing Whole Structures, Initialization of Structures.		
#Exemplar/Case Studies	Tower of Hanoi, Generation of Monthly balance sheet	

List of Laboratory Experiments/Assignments (Any 6 to 8 laboratory assignments) based on Programming	
1	To accept the number and Compute a) square root of number, b) Square of number, c) Cube of number d) check for prime, d) factorial of number e) prime factors.
2	To accept from user the number of Fibonacci numbers to be generated and print the Fibonacci series.
3	To accept an object mass in kilograms and velocity in meters per second and display its Momentum. Momentum is calculated as $e=mc^2$ where m is the mass of the object and c is its velocity.

4	In array do the following: 1. Find given element in array 2. Find Max and Min element 3. Find frequency of given element in array 4. Find Average of elements in Array.
5	Write a C program for employee salary calculation given, Basic, H.R.A. 20 % of Basic and D.A. 150 % of Basic.
6	To accept a student's marks for five subjects, compute his/her result. Student is passing if he/she scores marks equal to and above 40 in each course. If student scores aggregate greater than 75%, then the grade is distinguished. If aggregate is 60>= and <75 then the Grade of first division. If aggregate is 50>= and <60, then the grade is second division. If aggregate is 40>= and <50, then the grade is third division.
7	To accept two numbers from user and compute smallest divisor and Greatest Common Divisor of these two numbers.
8	Write a C program that accepts a string from the user and performs the following string operations- i. Calculate length of string ii. String reversal iii. Equality check of two Strings iii. Check palindrome ii. Check substring
9	Create Structure EMPLOYEE for storing details (Name, Designation, gender, Date of Joining and Salary), and store the data and update the data in structure.
10	Create class STORE to keep track of Products (Product Code, Name and price). Display menu of all products to users. Generate bills as per order.
Mini-Projects	
1	Calculator with basic functions. Add more functionality such as graphic user interface and Complex calculations.
2	Program that simulates rolling dice. When the program runs, it will randomly choose a number between 1 and 6 (Or other integer you prefer). Print that number. Request user to roll again. Set the min and max number that dice can show. For the average die, that means a minimum of 1 and a maximum of 6.
3	Guess Number: Randomly generate a number unknown to the user. The user needs to guess what that number is. If the user's guess is wrong, the program should return some sort of indication as to how wrong (e.g. the number is too high or too low). If the user guesses correctly, a positive indication should appear. Write functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers.
4	To calculate the salary of an employee given his basic pay (take as input from user). Calculate gross salary of employee. Let HRA be 10 % of basic pay and TA be 5% of basic pay. Let employees pay professional tax as 2% of total salary. Calculate net salary payable after deductions.
Learning Resources for Practical	
Text Books: Programming in ANSIC, 8e –E. Balagurusamy	

Reference Books:

1. B. S. Gottfried, Programming with C (Schaum's Outline Series), 2nd ed. McGraw-Hill, 1996.
2. S. C. Kochan, Programming in C, Sams Publishing, 3rd ed. 2004.
3. B. W. Kernighan and D. M. Ritchie, The C Programming Language, 2 nd ed. UK: Prentice Hall, 1988.
4. W. Kernighan and B. Pike, The Practice of Programming, UK: Addison-Wesley, 1999
5. H. M. Deitel and P. J. Deitel, C: How to program, 8 th ed. Pearson Education, 2015.
6. P. Prinz & T. Crawford, C in a Nutshell: The Definitive Reference, 2nd ed., O'Reilly Media, 2016

e-Books:https://studylib.net/doc/25796931/programming-in-ansic--8e---balagurusamy

MOOC / NPTEL/YouTube Links: https://onlinecourses.nptel.ac.in/noc22_cs40/preview
: https://onlinecourses.nptel.ac.in/noc23_cs53/preview

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a hands-on resource and reference. The instructor's manual needs to include prologue (about University/program/ institute/ department/foreword/ preface etc), copy of curriculum, conduction & Assessment guidelines, topics under consideration- concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Lab Journal

The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept in brief, features of tool/framework/language used, Design, test cases, conclusion. Program codes with sample output of all performed assignments are to be submitted as softcopy.

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journals may be avoided. Use of Drive containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Lab /TW Assessment

Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Laboratory Conduction

List of laboratory assignments is provided below for reference. The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy should address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute them among batches of students. It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of coding style, proper indentation and comments.

Use of open source software and recent versions is to be encouraged.

In addition to these, instructors may assign one real life application in the form of a mini-project.

based on the concepts learned. Instructors may also set one assignment or mini-project that is suitable to each branch beyond the scope of the syllabus.