Title: Structured Programming (Sessional)

Credits: 1.5 (2 hour 30 minutes per week)

Course Teacher:

Dr. Md. Iqbal Hasan Sarker, Assistant Professor, Dept. of CSE, CUET.
Omar Sharif, Lecturer, Dept. of CSE, CUET
Md. Billal Hossain, Lecturer, Dept. of CSE, CUET

Learning Resources:

Book-1: Programming in ANSI C
- E. Balagurusamy (6th Edition).

Book-2: Teach Yourself C
- Herbert Schildt (3rd Edition).

Catalog Description:

Structured Programming Language: Introduction: data types, operators, expressions; Input and output: standard input and output, formatted input and output; Control structures: branching, looping; Arrays: 1-D array, multidimensional array; Strings; Functions and program structure: parameter passing conventions, scope rules and storage classes, recursion; User defined data types: structures, unions, enumerations; File management; Error handling; Variable length argument list; Command line parameters; Header files; Preprocessor; Linking; Library functions.

Reference language: C

Prerequisite(s): None

Course Designation as Elective or Required: Required

Course Objectives:

1. Familiarization of Integrated Development Environment (IDE) and learn about it's different components.
2. Implement different programs using control structure, looping and functions.
3. Apply appropriate programming paradigm to solve problems by designing and debugging.
4. Analyze a large problem and break it into smaller parts, design each part as a module or function and write the solution.

Course Learning Outcomes: After successfully completing the course with a grade of D (2.0/4.0) or higher, the students should be able to do the following:

No	Course Learning Outcomes (CLOs)
1	Identify the steps involved in creating a program by control structure, looping and different programming techniques.
2	Analyze the problem and choose an appropriate programming paradigm to solve the problem.
3	Break a large problem into smaller parts, design each part as a module or function and develop the solution.

Mapping of Program Outcomes Addressed by CLOs:

CLO-PO mapping

CLO#	Program Outcome (PO)	PO#
1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.	1
2	**Problem Analysis:** Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principle of mathematics, natural sciences and engineering sciences.	2
3	**Design/Development of solutions:** Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations	3

CLO—PO Mapping:

Title of Course	**Programs Outcomes (PO)**												Complex Engineering Problem Solving	Complex Engineering Activities
	Engineering Knowledge	Problem Analysis	Design	Investigation	Modern Tools	Engineers and Society	Environment and sustainability	Ethics	Teamwork	Communication	Project Management and Finance	Life-long Learning		
CSE-142: Structured Programming (Sessional)	×	×	×										×	

Lesson Plan

With

Lesson Learning Outcomes (LLOs)

Week No.	Topic	Lesson Learning Outcome (at the end of the lesson students will be able to…)	Teaching-Learning Methodology	Assessment Method
Week-01	Introduction to Structured Programming	• Install and write programs on an IDE • Know about the basic structure of a simple C program • Write a simple C program	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-02	Study the concepts of constants, variables and data types.	• Know about constants, variables and data types • Declare variables • Implement these ideas in a program	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-03	Managing input-output operations	• Apply scanf and printf function • Implement Arithmetic, Relational, Logical, Assignment, Increment/Decrement, Bitwise operators in programs. • Use type conversion rule	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-04	Conditional logic, operators and expressions.	• Explain conditional logic. • Evaluate different types of expressions • Apply conditional logic to make decisions	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-05	Nested if-else, switch state-ment and conditional operator	• Implement programs using nested if-else • Explain the structure of switch statement • Apply conditional operator for making two-way decisions	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-06	Decision making and looping	• Explain the structure of while, do-while and for loop • Differentiate between counter-controlled and sentinel-controlled loop • Choose among while, do-while and for loop	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report

Week-07	Nested loops, Jump in loops	• Apply nested-loop for solving complex decision problems • Construct different nested loop structure such as for-while, while-for, for-for etc. • Use break and continue to skip part of a loops	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-08	Array, Searching and Sorting	• Declare and initialize arrays • Differentiate between runtime and compile time initialization of arrays • Apply searching and sorting techniques in array • Compute the complexity of searching and sorting	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-09	Multidimensional array and Character array	• Solve problems using multidimensional arrays • Declare and initialize string variables • Apply string-handling functions	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-10	Functions	• Define the structure of functions • Classify functions based on their arguments and return types • Describe the concept of scope, visibility and lifetime of variables in functions	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-11	Recursion	• Explain the concept and importance of recursion • Apply nesting of functions • Solve complex problems using recursion	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-12	Structure and File handling	• Implement structure in a program • Describe the process of opening and closing a file • Perform input, output and error handling operations in a file	Lecture on theoretical background, Implementation of programs	Lab performance, Lab report
Week-13	Quiz, Viva, Performance Test			