

Course Outline (OBE Format): Structured Programming (C)

Course Code: CSE-1102

Credit: 3.0

Level: 1st Year, 1st Semester

Course Teacher: Dr. Mahmudul Hasan,

Associate Professor, Dept. of CSE, Comilla University

1. Course Description

This course introduces structured programming concepts using C, focusing on algorithmic thinking, modular design, and efficient implementation. Emphasis is given to control structures, functions, arrays, pointers, and basic data handling techniques to build a strong programming foundation.

2. Course Objectives (COB – Course Objectives)

By the end of the course, students will:

- **COB1:** Understand fundamental programming concepts and problem-solving techniques
 - **COB2:** Develop structured programs using control statements and modular design
 - **COB3:** Analyze problems and implement solutions using arrays, strings, and functions
 - **COB4:** Apply basic memory concepts using pointers and structures
-

3. Course Learning Outcomes (CLOs)

CLO Code	Course Learning Outcome (CLO)	Bloom's Level
CLO1	Explain basic programming concepts, syntax, and execution process of C	Understand (C2)

CLO Code	Course Learning Outcome (CLO)	Bloom's Level
CLO2	Apply operators, conditional statements, and loops to solve problems	Apply (C3)
CLO3	Design and implement modular programs using functions and recursion	Apply (C3)
CLO4	Solve computational problems using arrays and strings	Apply (C3)
CLO5	Demonstrate basic use of pointers, structures, and file handling	Analyze (C4)

4. Program Learning Outcomes (PLOs)

(Aligned with standard engineering accreditation frameworks)

PLO Code	Description
PLO1	Engineering knowledge
PLO2	Problem analysis
PLO3	Design/development of solutions
PLO4	Investigation
PLO5	Modern tool usage
PLO6	Engineer and society
PLO7	Environment and sustainability
PLO8	Ethics
PLO9	Individual and teamwork
PLO10	Communication
PLO11	Project management
PLO12	Lifelong learning

5. CLO–PLO Mapping Matrix

CLO \ PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO9	PLO10	PLO12
CLO1	✓ (High)	✓ (Low)			✓ (Low)		✓ (Low)	✓ (Low)
CLO2	✓ (High)	✓ (High)	✓ (Medium)		✓ (Medium)	✓ (Low)		
CLO3	✓ (Medium)	✓ (High)	✓ (High)		✓ (Medium)	✓ (Low)		
CLO4	✓ (Medium)	✓ (High)	✓ (Medium)	✓ (Low)	✓ (Medium)			
CLO5	✓ (Medium)	✓ (Medium)	✓ (Medium)	✓ (Medium)	✓ (High)		✓ (Low)	✓ (Medium)

Legend:

- High (H) = Strong contribution
- Medium (M) = Moderate contribution
- Low (L) = Basic contribution

6. Teaching–Learning Strategy (OBE-Aligned)

Method

Purpose

Interactive lectures

Conceptual understanding (CLO1)

Live coding sessions

Skill development (CLO2–CLO4)

Problem-solving exercises

Analytical thinking (CLO2–CLO4)

Assignments/projects

Application and design (CLO3–CLO5)

Lab integration (optional)

Practical reinforcement

7. Assessment Strategy and CLO Mapping

Assessment Tool	CLOs Covered	Weight (%)
Class Tests/Quizzes/Attendance	CLO1, CLO2	10%
Assignments/Presentation	CLO2, CLO3, CLO4	10%
Midterm Exam	CLO1, CLO2, CLO3	20%
Final Exam	CLO2, CLO3, CLO4, CLO5	60%

8. CLO Attainment Measurement

CLO	Direct Assessment Tools	Target Attainment
CLO1	Quiz, Midterm	$\geq 60\%$ students score $\geq 50\%$
CLO2	Quiz, Assignment, Final	$\geq 60\%$ students score $\geq 50\%$
CLO3	Assignment, Midterm, Final	$\geq 60\%$ students score $\geq 50\%$
CLO4	Assignment, Final	$\geq 60\%$ students score $\geq 50\%$
CLO5	Final Exam	$\geq 60\%$ students score $\geq 50\%$

9. Weekly Topic vs CLO Mapping

Week	Topics	Mapped CLOs
1	Introduction to C	CLO1
2–3	Data types, operators	CLO1, CLO2
4	Conditional statements	CLO2
5–6	Loops	CLO2
7	Midterm	—
8	Functions, recursion	CLO3

9	Arrays	CLO4
10	Strings	CLO4
11	Pointers	CLO5
12	Structures	CLO5
13	File handling	CLO5
14	Review	All CLOs

10. Continuous Quality Improvement (CQI) Plan

- Analyze CLO attainment after each semester
 - Identify weak areas (e.g., recursion, pointers)
 - Introduce remedial classes and coding clinics
 - Update assignments with real-world problems
 - Integrate mini-projects for better design skills
-

11. Recommended Textbooks

- Brian W. Kernighan & Dennis M. Ritchie – *The C Programming Language*
 - Byron Gottfried – *Programming with C*
 - E. Balagurusamy – *Programming in ANSI C*
-

12. Graduate Attributes Developed

- Logical reasoning and structured thinking
- Problem-solving discipline
- Code readability and modular design mindset
- Foundation for advanced computing courses