Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

C Language Course Outline

Course Details

This course introduces students to the foundational concepts of computer programming using the C language. It is designed for beginners who have little to no prior coding experience. Ideal for students, professionals, or anyone interested in starting their journey into programming and software development. No prior programming knowledge is required.

Duration: 36 Hours
Class Frequency: 3 Classes per Week (2 Hours Each)
Delivery Mode: On-Campus

Course Objectives

By the end of this course, students will:

- Understand the fundamentals of structured programming using C.
- Write, compile, and debug simple C programs.
- Gain proficiency in using data types, variables, operators, and control structures.
- Apply functions, arrays, pointers, and strings in problem-solving.
- Develop structured and modular programming skills.
- Work on a final project applying all learned concepts.

Week-wise Course Plan

Week	Topics Covered	Learning Objectives	Assignments
1	- History & importance of C - Setting up compiler (Turbo C / GCC / IDEs) - Writing first program (Hello World) - Compilation & execution process	- Familiarize with C environment - Understand program structure	Write a “Hello World” and basic arithmetic program

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Week	Topics Covered	Learning Objectives	Assignments
2	- Keywords & identifiers - Constants & variables - Data types (int, float, char) - Type conversion & casting - Operators (arithmetic, relational, logical, assignment)	- Understand variable declaration & scope - Apply operators in expressions	Create a calculator program using operators
3	- `printf()`, `scanf()` - Decision making (`if`, `else if`, `switch`) - Looping (`for`, `while`, `do-while`)	- Perform input & output operations - Use control structures for logical flow	Write programs for multiplication table & menu-driven calculator
4	- Defining & calling functions - Function arguments (by value & reference) - Scope & lifetime of variables	- Break problems into modular functions	Program to calculate factorial & Fibonacci using functions
5	- 1D & 2D arrays - String handling (`gets()`, `puts()`, `strlen()`, `strcpy()`, etc.)	- Work with arrays for data storage - Manipulate strings in C	Program for matrix addition & string palindrome checker
6	- Introduction to pointers - Pointer arithmetic - Pointers with arrays & functions	- Understand memory management basics	Swap two numbers using pointers
7	- Structures & unions - File operations (create, read, write, append)	- Store and manage complex data - Perform basic file I/O operations	Program to store student records in a file
8	- Revision of core concepts - Final project development & presentation	- Apply all concepts in a real-world program	Final project submission

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Assessment Structure

- **Weekly Assignments:** 30%
- **Mid-Course Practical:** 20%
- **Final Project (End of Course):** 30%
- **Class Participation:** 20%

Final Project

Students will work on a project such as:

- Student Management System
- Library Record System
- Simple Banking System
- Quiz/Test Application

Recommended Resources

- *Let Us C* by Yashavant Kanetkar
- *The C Programming Language* by Brian W. Kernighan & Dennis M. Ritchie
- Online Tutorials (GeeksforGeeks, W3Schools, Programiz)
- IDEs: Code::Blocks, Dev-C++, Turbo C++, GCC
- W3Schools, MDN Web Docs,

Attendance Policy

Regular attendance is essential for successful course completion. Students are expected to attend at least **80% of classes**. More than **20% unexcused absences** may result in disqualification from the final project and certification.

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*