Course Title: C Programming

Course Description: This course provides a comprehensive introduction to the C programming language. Students will learn the basics of programming in C, including variables, data types, control structures, functions, arrays, pointers, and file handling. The course covers both theoretical concepts and practical programming exercises to reinforce learning.

Prerequisites: No prior programming experience is required. Basic understanding of computer operation and logic is beneficial.

Course Objectives:

1. Understand the fundamentals of the C programming language.
2. Learn how to write, compile, and execute C programs.
3. Gain proficiency in programming constructs such as variables, control structures, and functions.
4. Develop skills in working with arrays, pointers, and memory management in C.
5. Explore file handling and basic data structures in C.

Course Outline:

Module 1: Introduction to C Programming

- History and features of C programming language
- Setting up development environment (IDE, compiler)
- Writing and executing a simple C program

Module 2: Variables and Data Types

- Declaring variables and data types in C
- Basic input/output operations (printf, scanf)
- Constants and literals in C

Module 3: Control Structures

- Conditional statements (if-else, switch-case)
- Looping statements (for, while, do-while)
- Nested loops and loop control statements

Module 4: Functions

- Declaring and defining functions in C

- Function prototypes and function calls
- Passing arguments to functions and returning values

Module 5: Arrays and Strings

- Declaring and initializing arrays
- Accessing array elements and array manipulation
- Working with strings in C (character arrays)

Module 6: Pointers and Dynamic Memory Allocation

- Introduction to pointers and memory addresses
- Pointer arithmetic and pointer manipulation
- Dynamic memory allocation (malloc, calloc, realloc, free)

Module 7: Structures and Unions

- Defining and accessing structures in C
- Nested structures and structure pointers
- Introduction to unions and their usage

Module 8: File Handling

- Working with files in C (opening, reading, writing, closing)
- File operations using standard library functions (fopen, fread, fwrite)
- Error handling and file management in C programs

Module 9: Advanced Topics

- Preprocessor directives and macros
- Bitwise operations and bitwise manipulation
- Recursion and recursive functions in C

Module 10: Introduction to Data Structures

- Overview of basic data structures (arrays, linked lists, stacks, queues)
- Implementing basic data structures in C
- Introduction to algorithms and algorithmic complexity

Module 11: Real-world Projects and Case Studies

- Working on real-world projects and case studies
- Designing and implementing end-to-end C programs

- Presenting findings and insights from projects

Module 12: Capstone Project

- Developing a comprehensive C program
- Identifying a problem or scenario
- Designing and implementing a solution using skills learned throughout the course

Assessment:

- Weekly programming assignments to reinforce learning concepts.
- Midterm project: Developing a basic C program focusing on fundamental concepts (variables, control structures, functions).
- Final project: Designing and implementing a comprehensive C program addressing a real-world problem or scenario.

Textbook: "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie

Additional Resources:

- Online tutorials and documentation (C programming resources, C programming forums, etc.).
- Supplemental readings and materials provided by the instructor.

Grading:

- Assignments: 30%
- Midterm Project: 20%
- Final Project: 40%
- Participation and Attendance: 10%

Attendance Policy: Regular attendance is expected. Students are allowed a maximum of three unexcused absences. Excessive absences may result in a reduction of the final grade.

Office Hours: Instructor office hours will be held twice a week for additional help and clarification.