Course content

The course introduces the principles of programming and the foundational concepts of structured programming using C language. It is designed to develop students' ability to use C as a tool for software development and problem-solving in computer science. The course emphasizes hands-on programming practices and prepares students for advanced courses and future roles in software and hardware development.

The following topics are covered in the course:

1. Programming Basics: Fundamental principles and methods of structured programming, including program design and debugging.
2. C Language Syntax and Semantics: Understanding the core syntax, semantics, and usage of the C programming language.
3. Data Structures and Algorithms: Implementing basic algorithms and understanding data representation in C.
4. Problem Solving with C: Writing and optimizing C programs to solve computational problems.
5. Code Analysis and Debugging: Techniques for identifying critical algorithms and debugging processes to ensure program correctness.
6. Foundational Software Development: Building a foundation for future software development courses and applications in various programming languages.

Course objectives

Knowledge

1. Understand the fundamental principles of programming, structured programming, and problem-solving with C language.
2. Learn the syntax and semantics of the C programming language and its application in software development, including implementation strategies, syntax, memory management, scope rules, control structures, types, language paradigms, modules, semantics, domain-specific languages.
3. Grasp the basics of data structures and algorithms in the context of C programming.

Skills

1. Write, debug, and optimize C programs to solve computational and real-world problems.
2. Use pseudocode and C language to describe and solve problems effectively.
3. Analyze programming requirements and propose solutions using fundamental algorithms and programming concepts.
4. Compare, synthesize, and optimize algorithms and programs based on design principles.

Competencies

1. Apply programming principles to identify and solve problems, focusing on key algorithms and critical code components.
2. Translate problem requirements into correct and efficient C programs through iterative debugging and validation.

3. Build a solid foundation for advanced programming courses and the use of other programming languages in professional development.