Course Content

This course is a foundational subject for students majoring in Computer Science, focusing on essential programming skills. These skills are crucial not only for mastering subsequent professional courses but also for future employment in computer and software-related fields. Students are required to understand the fundamental principles, methods, and techniques of programming and demonstrate proficiency through hands-on practice and a programming level assessment.

The following topics are covered in the course:

1. Programming Principles and Methods: Understanding fundamental programming concepts, algorithms, and methodologies for solving computational problems.
2. Problem-Solving and Algorithm Design: Developing and designing types, functions, or algorithms to meet specific requirements for solving complex problems.
3. Innovation in Programming: Encouraging creativity and innovation in program design while considering social, environmental, and other broader factors.
4. Experimental Design and Implementation: Designing experiments and constructing programming environments to solve targeted problems effectively.
5. Debugging and Tools Utilization: Learning the principles and limitations of common debugging tools and simulation software for problem-solving.
6. Analysis and Conclusion: Analyzing experimental results and synthesizing information to draw rational and effective conclusions.

Course Objectives

Knowledge

1. Gain a solid understanding of programming principles and methodologies for problem-solving.
2. Learn the fundamentals of debugging tools and simulation software.
3. Understand the limitations of modern software tools in solving computational problems.

Skills

1. Design and implement algorithms, types, and functions tailored to solve specific and complex problems.
2. Conduct experiments by constructing programming environments and collecting data scientifically.
3. Analyze experimental data and derive reasonable conclusions through comprehensive information integration.
4. Utilize appropriate debugging and simulation tools to analyze and design solutions for complex problems.

Competencies

1. Demonstrate creativity and innovation in program design while considering broader social and environmental factors.
2. Evaluate and optimize problem-solving methods using modern tools.

3. Develop or select software tools to meet specific requirements, predict outcomes, and analyze their limitations.
4. Build a strong foundation for future studies and careers in software development and computational problem-solving.