

Software Development Course Modules

Course Overview

This course contents provides a comprehensive introduction to software development, covering the fundamental concepts, tools, and practices required to build modern software applications. Students will learn core programming skills, software design principles, and development methodologies. Through hands-on projects and collaborative exercises, participants will gain practical experience in writing, testing, and debugging code using popular programming languages and frameworks.

Learning Outcomes:

By the end of the course, students and professionals will be able to:

- Design and implement software applications using industry-standard languages.
- Apply key programming and debugging techniques to solve problems.
- Work collaboratively using version control and Agile practices.
- Understand the architecture and lifecycle of software systems.
- Build and deploy a simple full-stack application.

Prerequisites:

1. Basic proficiency in mathematics and familiarity with spreadsheets.
2. No prior programming experience required.

Course Modules

Modules	Title	Topics
1.	Introduction to Software Engineering	<ul style="list-style-type: none">➤ Definition and Scope➤ Software Characteristics and Applications
2.	Software Development Life Cycle (SDLC)	<ul style="list-style-type: none">➤ Waterfall Model➤ V-Model➤ Iterative, Incremental, Spiral Models➤ Agile Approaches
3.	Requirements Engineering	<ul style="list-style-type: none">➤ Types of Requirements (Functional & Non-functional)➤ Requirement Gathering and Elicitation Techniques

		<ul style="list-style-type: none"> ➤ Requirements Specification and Documentation
4.	Software Design	<ul style="list-style-type: none"> ➤ Design Principles and Patterns ➤ Architectural Design (MVC, Layered Architecture) ➤ Interface Design ➤ Component-level Design
5.	Object-Oriented Design and Analysis	<ul style="list-style-type: none"> ➤ UML Diagrams (Class, Sequence, Activity, etc.) ➤ Design Patterns (Singleton, Factory, Observer, etc.)
6.	Software Project Management	<ul style="list-style-type: none"> ➤ Project Planning and Scheduling ➤ Cost Estimation (COCOMO, Function Point) ➤ Risk Management ➤ Team Management and Communication
7.	Software Testing and Quality Assurance	<ul style="list-style-type: none"> ➤ Testing Levels (Unit, Integration, System, Acceptance) ➤ Black-box and White-box Testing ➤ Test Case Design ➤ Software Quality Standards (ISO, CMMI)
8.	Software Maintenance and Evolution	<ul style="list-style-type: none"> ➤ Types of Maintenance ➤ Re-engineering and Refactoring ➤ Change Management
9.	Configuration Management and Version Control	<ul style="list-style-type: none"> ➤ Git and GitHub Basics ➤ Change Control Processes ➤ Continuous Integration/Delivery (CI/CD)
10.	Software Metrics and Measurement	<ul style="list-style-type: none"> ➤ Code Quality Metrics ➤ Productivity Metrics ➤ Maintainability Index
11.	Ethics and Professional Practice knowledge	<ul style="list-style-type: none"> ➤ Intellectual Property ➤ Software Licensing (Open Source vs. Proprietary) ➤ Ethical Responsibilities of a Software Engineer
12.	Capstone Project or Case Study	<ul style="list-style-type: none"> ➤ Real-world Software Development Project ➤ Application of All Learned Concepts

		➤ Team Collaboration and Documentation
--	--	--