BTU | BUSINESS & TECHNOLOGY UNIVERSITY

SOFTWARE DEVELOPMENT

Software Development

Course Description

This course provides an introduction to the fundamentals of software development, covering topics such as programming languages and frameworks, software design and architecture, software testing and quality assurance, software project management, and software deployment and maintenance. Students will learn about object-oriented programming, design principles and patterns, project planning and estimation, software testing methodologies and frameworks, team collaboration and communication, software deployment strategies, software maintenance and support, software security and compliance, and emerging trends in software development such as AI/ML and blockchain. Practical assignments and projects will be provided to apply the concepts and techniques learned in each module.

Learning Outcomes

Upon completion of this course, students will be able to:

- Understand the fundamentals of software development, including programming languages and frameworks, software design and architecture, software testing and quality assurance, software project management, and software deployment and maintenance.
- Apply object-oriented programming concepts and techniques to software development projects.
- Design and implement software solutions using appropriate software design principles and patterns.
- Develop and implement appropriate software testing methodologies and frameworks to ensure software quality.
- Apply appropriate software project management methodologies and tools to manage software development projects.
- Implement appropriate software deployment strategies and maintenance procedures to ensure software reliability and security.
- Understand emerging trends in software development such as AI/ML and blockchain and their potential impact on software development.

Module 1: Introduction to Software Development

1.1 Overview of Software Development

1. Introduction to software development and its importance
2. Software development methodologies (e.g. Agile, Waterfall)
3. Overview of software development life cycle (SDLC)

1.2 Software Development Tools and Technologies

1. Overview of programming languages and frameworks
2. Integrated Development Environments (IDEs) and text editors
3. Version control systems and collaboration tools

1.3 Basics of Object-Oriented Programming (OOP)

1. Introduction to OOP concepts (e.g. classes, objects, inheritance, polymorphism)
2. Overview of popular OOP languages (e.g. Java, C#, Python)

Homework Assignment: Introduction to Programming

1. Choose a programming language (e.g. Java, Python, C#)
2. Write a simple program using the chosen language
3. Use an IDE or text editor to write and run the program
4. Implement appropriate variables, loops, and conditional statements in the program
5. Write a report summarizing the program and programming experience

Evaluation Criteria:

- Proper use of programming language syntax and constructs
- Proper execution and functionality of the program
- Clear and concise report summarizing the program and programming experience

Module 2: Software Design and Architecture

2.1 Software Design Principles

1. Overview of design patterns (e.g. MVC, Observer, Singleton)
2. SOLID principles of software design
3. Designing for scalability, maintainability, and extensibility

2.2 Software Architecture Patterns

1. Overview of software architecture patterns (e.g. client-server, microservices, event-driven)
2. Choosing the appropriate architecture pattern for the project requirements
3. Designing for security, performance, and reliability

2.3 Software Modeling and Diagramming

1. Overview of software modeling and diagramming tools (e.g. UML, ERD)
2. Creating appropriate models and diagrams for the project requirements
3. Documenting and communicating software design and architecture decisions

Homework Assignment: Software Design and Architecture

1. Choose a simple software project (e.g. a task management app, a simple game)
2. Develop a software design and architecture plan for the project
3. Implement the design and architecture plan in the project code
4. Test and validate the project's functionality and design decisions
5. Write a report summarizing the software design and architecture decisions and implementation

Evaluation Criteria:

- Adequate use of appropriate software design principles and architecture patterns
- Proper execution and functionality of the project
- Clear and concise report summarizing the software design and architecture decisions and implementation

Module 3: Software Testing and Quality Assurance

3.1 Software Testing Fundamentals

1. Overview of software testing types (e.g. unit, integration, system, acceptance)
2. Importance of software testing and quality assurance
3. Common software testing frameworks and tools

3.2 Test-Driven Development (TDD)

1. Introduction to TDD and its benefits
2. Implementing TDD in software development process
3. Common TDD frameworks and tools

3.3 Continuous Integration and Delivery (CI/CD)

1. Introduction to CI/CD and its benefits
2. Implementing CI/CD in software development process
3. Common CI/CD frameworks and tools

Homework Assignment: Software Testing and Quality Assurance

1. Choose a simple software project (e.g. a calculator app, a simple website)
2. Develop appropriate test cases for the project's functionality
3. Implement the tests using a testing framework (e-.g. JUnit, Selenium)
4. Use TDD to develop and validate the project functionality
5. Implement CI/CD pipeline for the project
6. Write a report summarizing the testing and quality assurance process and implementation

Evaluation Criteria:

- Adequate use of appropriate testing types and frameworks
- Proper execution and functionality of the project
- Clear and concise report summarizing the testing and quality assurance process and implementation

Module 4: Software Project Management

4.1 Project Planning and Estimation

1. Overview of project management methodologies (e.g. Agile, Scrum)
2. Creating project plans and estimates
3. Managing project scope and requirements

4.2 Software Project Development Process

1. Overview of software development methodologies (e.g. Waterfall, Agile)
2. Managing the software development process (e.g. requirements gathering, design, development, testing, deployment)

4.3 Team Collaboration and Communication

1. Effective team collaboration and communication strategies
2. Managing team conflicts and challenges
3. Implementing project management tools for team collaboration

Homework Assignment: Software Project Management

1. Choose a software project idea (e.g. a social media app, a game)
2. Develop a project plan and estimate for the project
3. Implement the project development process using an Agile methodology
4. Use appropriate project management tools for team collaboration and communication
5. Write a report summarizing the project management process and implementation

Evaluation Criteria:

- Adequate use of appropriate project management methodologies and tools
- Proper execution and functionality of the project
- Clear and concise report summarizing the project management process and implementation

Module 5: Software Deployment and Maintenance

5.1 Software Deployment Strategies

1. Overview of software deployment strategies (e.g. manual, automated)
2. Choosing the appropriate deployment strategy for the project
3. Implementing deployment automation tools and frameworks

5.2 Software Maintenance and Support

1. Handling software bugs and errors
2. Providing software support and customer service
3. Managing software updates and maintenance

5.3 Software Security and Compliance

1. Overview of software security risks and best practices
2. Ensuring software compliance with industry regulations and standards
3. Implementing security features and protocols in software development process

Homework Assignment: Software Deployment and Maintenance

1. Choose a simple software project (e.g. a blogging platform, an e-commerce website)
2. Implement appropriate deployment strategy and automation tools for the project
3. Implement software maintenance and support procedures for the project

4. Ensure software compliance with industry regulations and standards
5. Write a report summarizing the software deployment and maintenance process and implementation

Evaluation Criteria:

- Proper implementation of appropriate deployment strategy and automation tools
- Adequate handling of software bugs and errors, and provision of software support
- Proper implementation of security features and protocols in the software development process
- Clear and concise report summarizing the software deployment and maintenance process and implementation

Module 6: Advanced Topics in Software Development

6.1 Software Performance Optimization

1. Overview of software performance optimization techniques
2. Implementing software performance optimization techniques (e.g. code optimization, database optimization)
3. Monitoring and analyzing software performance metrics

6.2 Artificial Intelligence and Machine Learning in Software Development

1. Overview of AI and ML concepts and techniques
2. Implementing AI and ML in software development (e.g. natural language processing, image recognition)
3. Evaluating and validating AI and ML models

6.3 Emerging Trends in Software Development

1. Overview of emerging trends in software development (e.g. blockchain, serverless computing, quantum computing)
2. Understanding the potential impact of emerging trends on software development
3. Implementing emerging trends in software development

Homework Assignment: Advanced Topics in Software Development

1. Choose a software project idea that can benefit from AI/ML, software performance optimization, or emerging trends in software development

2. Implement appropriate techniques or trends in the project development process
3. Evaluate and validate the implementation of the techniques or trends
4. Write a report summarizing the implementation and evaluation process

Evaluation Criteria:

- Adequate use of appropriate techniques or trends in the project development process
- Proper execution and functionality of the project
- Clear and concise report summarizing the implementation and evaluation process

This course taught the fundamentals of software development, including programming languages and frameworks, software design and architecture, software testing and quality assurance, software project management, and software deployment and maintenance. It covered topics such as object-oriented programming, software design principles and patterns, project planning and estimation, software testing methodologies and frameworks, team collaboration and communication, software deployment strategies, software maintenance and support, software security and compliance, and emerging trends in software development such as AI/ML and blockchain. The course also included practical assignments and projects to apply the concepts and techniques learned in each module.