Java Programming & Software Engineering Fundamentals

Ngane Emmanuel

2024-11-06

Course Outline: Java Programming & Software Engineering Fundamentals

Module 1: Introduction to Java and Software Engineering

- **Objective:** Familiarize students with the Java programming environment and introduce software engineering concepts.
- Topics:
  - Overview of Java: JVM, JRE, JDK, installation, and setup.
  - Introduction to Integrated Development Environments (IDEs) – (e.g., Eclipse, IntelliJ, or VS Code).
  - What is Software Engineering? Overview of the software development lifecycle (SDLC).
  - Overview of Java application in the SDLC.
  - Project: Setting up a "Hello World" Java project, compiling, and running Java programs.

Module 2: Basic Programming Concepts in Java

- **Objective:** Cover the fundamental building blocks of Java programming.
- Topics:
  - Variables, data types, and operators.
  - Input and Output in Java.
  - Basic control structures (if-else, switch, loops).
  - Introduction to debugging and testing: Using print statements and basic IDE debuggers.
  - Software engineering basics: Requirements gathering for a small program (e.g., a calculator).
  - Project: Building a simple calculator or temperature converter.

Module 3: Object-Oriented Programming (OOP)

- **Objective:** Introduce and apply OOP principles in Java.
- Topics:
  - Classes and Objects: Understanding `class` structure and `object` instantiation.
  - Encapsulation, Inheritance, Polymorphism, and Abstraction.
  - Access modifiers (public, private, protected).
  - Constructors, getters, and setters.
  - Basic UML for software design: Class diagrams.
  - Software engineering: Introduction to design principles and why OOP is used in software design.
  - Project: Create a simple "library system" with classes for `Book`, `Library`, and `User`.

Module 4: Data Structures and Algorithms in Java

- **Objective:** Develop a foundation in data structures and introduce algorithmic thinking.
- Topics:
  - Arrays, Lists, and basic Collection Framework (ArrayList, LinkedList).
  - Introduction to sorting and searching algorithms (e.g., bubble sort, binary search).
  - Exception handling in Java.
  - Software engineering: Emphasis on efficiency and performance analysis.
  - Project: Extend the "library system" to include data storage and sorting of books by title, author, etc.

Module 5: Introduction to Software Development Life Cycle (SDLC)

- **Objective:** Explore the SDLC stages and integrate it with Java project development.
- Topics:
  - Overview of SDLC models (Waterfall, Agile, etc.).
  - Requirement gathering and functional vs. non-functional requirements.
  - Basic project planning and dividing tasks into phases.
  - Software testing and debugging techniques.
  - Project: Collect requirements, design, and implement a "To-Do List" application using a basic SDLC approach.

Module 6: Intermediate Java Programming Concepts

- **Objective:** Introduce intermediate Java concepts and build larger, more complex systems.
- Topics:
  - File I/O: Reading from and writing to files.
  - Java collections (HashMap, HashSet, TreeSet).
  - Multithreading and concurrency basics.
  - Introduction to JavaFX for GUI development.
  - Software engineering: Introduction to version control with Git.
  - Project: Build a simple address book with a GUI interface and file storage.

Module 7: Software Design and Architecture

- **Objective:** Teach the importance of software design and architecture in building scalable applications.
- Topics:
  - Understanding software architecture patterns (Layered, Client-Server, N-Tier).
  - Introduction to design patterns: Singleton, Factory, and Observer.
  - Refactoring code for maintainability.
  - Project: Apply a layered architecture to build a "simple e-commerce system" with classes for `Product`, `Order`, and `User`.

Module 8: Database Integration with Java

- **Objective:** Introduce relational databases and SQL, and connect Java applications to a database.
- Topics:
  - Introduction to databases (MySQL, SQLite) and SQL basics.
  - JDBC for database connectivity in Java.
  - Basic CRUD operations in Java using JDBC.
  - Software engineering: Database design principles and normalizing tables.
  - Project: Extend the e-commerce system to store data in a database.

Module 9: Advanced Software Engineering Concepts

- **Objective:** Explore advanced software engineering concepts such as testing, maintenance, and deployment.
- Topics:
  - Software testing: Unit testing with JUnit and basic test-driven development (TDD).
  - Code reviews and version control best practices.
  - Continuous integration/continuous deployment (CI/CD) basics.
  - Documentation and user guides for software.
  - Project: Write unit tests for the e-commerce application and practice CI/CD using GitHub Actions.

Module 10: Capstone Project

- **Objective:** Apply all the learned concepts to build a complete application.
- Topics:
  - Project planning, requirement gathering, and task breakdown.
  - Full SDLC application: Design, implementation, testing, and documentation.
  - Integrate all components (OOP, file I/O, database, GUI, design patterns).
  - Final Project: Build a "Student Management System" or a "Hotel Booking System" that incorporates all learned Java and software engineering skills.