

NATIONAL UNIVERSITY OF MODERN LANGUAGES
ISLAMABAD



Software Engineering

Assignment No: 01

Submitted to
Ms. Frnaz Akbar

Submitted By
Junaid Asif
(BSAI-144)

Submission Date: February 11, 2025

Software Engineering Principles

Software engineering principles are fundamental guidelines that help developers create high-quality, maintainable, and scalable software. Here are some key principles:

1. SOLID Principles (for Object-Oriented Design)

- **Single Responsibility Principle (SRP):** A class should have only one reason to change.
- **Open/Closed Principle (OCP):** Software entities should be open for extension but closed for modification.
- **Liskov Substitution Principle (LSP):** Subtypes must be substitutable for their base types.
- **Interface Segregation Principle (ISP):** Clients should not be forced to depend on interfaces they do not use.
- **Dependency Inversion Principle (DIP):** High-level modules should not depend on low-level modules; both should depend on abstractions.

2. DRY (Don't Repeat Yourself)

- Avoid duplication of code to reduce redundancy and improve maintainability.

3. KISS (Keep It Simple, Stupid)

- Write simple and clear code instead of over-engineering solutions.

4. YAGNI (You Ain't Gonna Need It)

- Do not add unnecessary functionality unless it is required.

5. Encapsulation

- Hide implementation details and expose only what is necessary.

6. Modularity

- Break down software into smaller, independent modules for better maintainability.

7. Separation of Concerns (SoC)

- Divide a system into distinct sections, each handling a specific responsibility (e.g., MVC pattern).

8. Testability

- Code should be written in a way that makes unit testing and integration testing easier.

9. Code Reusability

- Design software components so they can be reused in different applications.

10. Scalability

- Ensure the system can handle growth without major redesigns.

