Introduction to Object Oriented Programming

by Dr Reema Rallan

Which is better to modify and why?

What is Object-Oriented Programming?

OOP uses objects to represent and manipulate data.

Objects combine data (attributes) and methods (actions).

Classes are blueprints for creating objects.

OOP prioritizes data and interactions over functions.

Promotes flexibility and code reuse in complex systems.

Basic OOP Concepts

1 **Classes**

Blueprints for objects, defining attributes and methods.

2 **Objects**

Instances of classes, with their own data and methods.

3 **Inheritance**

Objects inherit properties and methods from parent classes.

4 **Encapsulation**

Data is hidden within objects, accessed through methods.

5 **Polymorphism**

Objects of different classes can respond differently to the same method.

6 **Data Abstraction**

Hiding unnecessary implementation details from users.

Core Concepts of OOP

Encapsulation: Bundling data and methods in objects.

Inheritance: Creating new classes from existing ones.

Polymorphism: Flexible behavior using parent and child classes.

Abstraction: Hiding details, focusing on what matters.

These principles together make OOP robust and efficient.

Encapsulation

Protects data within objects.

Controlled access using access modifiers (e.g., private, public).

Enables modular design for easy updates.

Simplifies testing and debugging.

Inheritance

- Reuse existing classes to create new ones.
- Reduces redundancy, saves time.
- Enables hierarchical classification in systems.
- Allows method overriding for customization.

Polymorphism

- Treat objects as instances of their parent class.
- Supports method overriding and dynamic behavior.
- Enables flexible and reusable code.
- Simplifies extensibility for future features.

Abstraction

- Hides complex implementation details.
- Focuses on essential functionalities only.
- Encourages clean interfaces for interaction.
- Makes large systems manageable.

Benefits of Object Oriented Programming

1

Modularity

Breaking down complex programs into smaller, manageable units.

2

Reusability

Using existing code in multiple projects, saving time and effort.

3

Maintainability

Easier to modify and update code without impacting other parts.

4

Scalability

Adapting to changing requirements and growing project size.

C vs. C++

C

A procedural programming language, known for its low-level access and efficiency.

C++

An object-oriented extension of C, providing powerful features like classes, inheritance, and polymorphism.