An Overview of Java Programming

Dr. Partha Pratim Sarangi
School of Computer Engineering

Object-Oriented Programming

- All computer programs consists of two elements: code and data.
- A computer program can be conceptually organized around its code or around its data.
- Based on this principle computer programs can be written by using two models: Process-oriented model and Object-oriented model.
- The process-oriented model can be thought of as code acting on data.
- The problems with this approach appear as programs grow larger and more complex.
- To manage complexity and redundancy, the second approach, called object-oriented model, was conceived.

Contd…

- Object-oriented programming organizes a program around its data (that is objects) and a set of well-defined interfaces to that data.
- An object is a self-contained unit that bundles or packages both data (attributes) and the operations (bihavior).
- Object-Oriented Programming (OOP) is a methodology to design a program using classes and objects.
- Objects are instances of classes and classes serve as blueprints for creating objects.
- Each object has attributes (data) and methods (functions or procedures) that operate on that data.
- Object-oriented programming (OOP) is at the core of Java.

Object-Oriented Programming overview

- All object-oriented programming languages provide mechanisms that help to implement the object-oriented model.
- There are four key principles of object-oriented programming such as abstraction, encapsulation, inheritance, and polymorphism.
- **Abstraction** is the process of simplifying complex systems by modeling classes based on the essential properties and behaviors.
- It involves focusing on the essential characteristics of an object while ignoring the non-essential details.

- Abstraction helps in managing complexity by providing a clear separation between what an object does and how it achieves that functionality.
- It allows developers to create abstract models that represent real-world entities or concepts, with a focus on relevant details.
- In Java, abstraction is often achieved using abstract classes and interfaces.
- Abstract classes provide a way to define common characteristics and some common behavior, leaving the specific implementation details to the derived classes.
- Interfaces define a contract for classes to implement, specifying a set of methods without providing the implementation.

Encapsulation

- Encapsulation is the bundling of data (attributes) and methods (functions) that operate on the data into a single unit, known as a class.
- It involves restricting direct access to some of an object's components and only exposing what is necessary through well-defined interfaces.
- Encapsulation helps in hiding the internal details of an object and exposing only what is necessary for other parts of the program to interact with it.
- It provides data security by controlling access to an object's attributes, ensuring that they are accessed and modified in a controlled manner.

- In Java, encapsulation is achieved through access modifiers (e.g., private, protected, public) and by providing getter and setter methods to access and modify the object's attributes.
- By making attributes private and providing public methods for access and modification, encapsulation enforces controlled and safe interaction with the object.
- Both abstraction and encapsulation are complementary principles in OOP. Abstraction helps in creating models, while encapsulation helps in implementing those models by bundling data and methods and controlling access to them.

Inheritance

- Inheritance is a mechanism that allows a class (subclass or derived class) to inherit the properties and behaviors of another class (superclass or base class). It enables code reuse and the creation of a hierarchy of classes.
- Inheritance promotes code reuse by allowing a new class to take on the attributes and behaviors of an existing class, reducing redundancy and promoting a hierarchical organization of code.
- Implementation: Inheritance is implemented using the extends keyword in languages like Java. The subclass inherits fields and methods from the superclass.

Polymorphism:

- Polymorphism means "many forms." It allows objects of different types to be treated as objects of a common type. Polymorphism can be achieved through method overloading and method overriding.
- Polymorphism promotes flexibility and extensibility in code. It allows methods to be applied to objects of various types, simplifying code and making it more adaptable to changes.
- Polymorphism is implemented through method overloading (having multiple methods with the same name but different parameter lists) and method overriding (redefining a method in a subclass with the same signature as in the superclass).

Summary

- These four principles, encapsulation, abstraction, inheritance, and polymorphism are referred to as four pillar of object-oriented programming and form the foundation of object-oriented design.
- They contribute to building software that is modular, reusable, and easier to understand and maintain.