Lecture # 3

Imran Ahsan

Outline

- Encapsulation
- Polymorphism in Java
- Inheritance in Java
- Abstraction in Java
- Interfaces

Encapsulation

- **Encapsulation in java** is a *process of wrapping code and data together into a single unit*, for example capsule i.e. mixed of several medicines.
- We can create a fully encapsulated class in java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.

Advantage of Encapsulation in java

- By providing only setter or getter method, you can make the class **read-only or write-only**.
- It provides you the **control over the data**. Suppose you want to set the value of id i.e. greater than 100 only, you can write the logic inside the setter method.

Inheritance

- **Inheritance in java** is a mechanism in which one object acquires all the properties and behaviors of parent object.
- Inheritance represents the **IS-A relationship**, also known as *parent-child* relationship.

Why use inheritance in java

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Types of Inheritance

- 1) Single
- 2) Multilevel
- 3) Hierarchal
- 4) Multiple
- 5) Hybrid

Abstraction

- **Abstraction** is a process of hiding the implementation details and showing only functionality to the user.
- Another way, it shows only important things to the user and hides the internal details for example sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery.
- Abstraction lets you focus on what the object does instead of how it does it.

Abstraction

- A class that is declared as abstract is known as **abstract class**. It needs to be extended and its method implemented. It cannot be instantiated.

abstract class A{}

abstract method

A method that is declared as abstract and does not have implementation is known as abstract method.

abstract void printStatus();//no body and abstract

Interface

- An **interface in java** is a blueprint of a class. It has static constants and abstract methods.
- The interface in java is **a mechanism to achieve abstraction**. There can be only abstract methods in the java interface not method body. It is used to achieve abstraction and multiple inheritance in Java.

Why use Java interface?

There are mainly three reasons to use interface:

- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.

- class – class extends
- class – interface implements
- interface – interface extends

Abstract Class vs Interface

Abstract Class	Interface
• Abstract class can have abstract and non-abstract methods. • Abstract class doesn't support multiple inheritance. • Abstract class can have final, non-final, static and non-static variables. • Abstract class can provide the implementation of interface. • The abstract keyword is used to declare abstract class.	• Interface can have only abstract methods. • Interface supports multiple inheritance. • Interface has only static and final variables. • Interface can't provide the implementation of abstract class. • The interface keyword is used to declare interface.

End.