- A very important fact to remember is **that Java does not support multiple inheritance**. This means that a class cannot extend more than one class. **However, a class can implement one or more interfaces**, which has helped Java get rid of the impossibility of multiple inheritance.
- If a class inherits a method from its superclass, then there is a chance to override the method provided that it is not marked final. Overriding means to override the functionality of an existing method.
- **Polymorphism** is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object. Any Java object that can pass more than one IS-A test is considered to be polymorphic.

  The algorithms that are said to be polymorphic: that is, the same method can be used on many different implementations of the appropriate collection interface.
- Likewise in Object-oriented programming, abstraction is a process of hiding the implementation details from the user. In Java, abstraction is achieved using Abstract classes and interfaces.

  Abstract classes may or may not contain *abstract methods*, i.e., methods without body

  But, if a class has at least one abstract method, then the class **must** be declared abstract

  If a class is declared abstract, it cannot be instantiated.

  To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.

- **Encapsulation** in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as **data hiding**.

To achieve encapsulation in Java –

Declare the variables of a class as private.

Provide public setter and getter methods to modify and view the variables values.

- An **interface** is a reference type in Java. It is **a collection of abstract methods**. A class implements an interface, thereby inheriting the abstract methods of the interface. an interface may also contain constants, default methods, static methods, and nested types. A class describes the attributes and behaviors of an object. And an interface contains behaviors that a class implements. Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class. You cannot instantiate an interface. All of the methods in an interface are abstract. An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final. An interface can extend multiple interfaces

- Java **Generic** methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods, or with a single class declaration, a set of related types, respectively.