Chapter 8: Java Abstract Classes & Interfaces

Abstract Classes:

- Abstract class is class which may contain one or more abstract method.
- Abstract method is a method whose actions have to be redefined by the subclasses.
- Abstract classes are used to provide a template or design for concrete subclasses down the inheritance tree.

Rules for using Abstract Classes:

- We must override all abstract methods of abstract class in concrete subclasses.
- We cannot create objects of abstract class i.e. abstract class cannot be instantiated.
- Reference of abstract class can be used to refer to the objects of its subclasses i.e. polymorphic references.
- It is illegal to declare an abstract method in a class which is not declared abstract.
- An abstract class can be declared without abstract method.
- A method cannot be declared as both abstract and final.
- A method cannot be declared as both abstract and private.
- If the subclass is abstract, then it is not mandatory to implement all abstract methods of a superclass.

Interface:

- An Interface is a class that contains methods which are all abstract.

Rules for using Interfaces:

- All methods in interface are implicitly public and abstract. They cannot be private or protected.
- Interface methods must not be static or final.
- Interface abstract methods cannot have body.
- An interface declares only constants and not instance variables. All variables declared in an interface are public, final and static.
- Interface cannot be instantiated.
- When implementing interface the method of interface should have access specifier public.

	Abstract Classes	Interfaces
Methods	Abstract class can contain some abstract methods and some concrete methods.	Interface can contain only abstract methods.
Variables	Abstract class can contain instance variables also.	Interface cannot contain instance variables. It contains only constants.
Constructors	Abstract classes can have constructors, and those constructors are always called when a concrete subclass is instantiated.	Interfaces do not have constructors.
Multiple Inheritance	Abstract classes does not support multiple inheritance	Interfaces support multiple inheritance
Changes	If a new method is added to an abstract class, then there is an option of providing implementation and therefore all existing code works without any change.	If a new method is added to an interface, then it is required to track down all the implementations of the interface and define it.
Usefulness	Abstract classes are useful in situation when some general methods should be implemented and specialization behavior should be implemented by the subclasses.	Interfaces are useful in a situation when all it methods need to be implemented by subclasses.