Abstract Classes

An abstract method is a one which does not provide an implementation (body).

- An abstract class is a class declared with the abstract keyword.
- A class which includes any abstract method must be declared as abstract.
- Non-abstract classes are referred to as concrete classes.
- A concrete subclass of an abstract class must provide an implementation for all the inherited abstract methods.

Why declare a class as abstract?

1. Code reuse: To provide a base that models the common features of several related classes. The implementation of the methods that are specific to the subclasses can be delayed. Several subclasses can extend this base.
2. Programming in the general: Abstract subclasses can be used in polymorphic processing of collection of objects of related classes.

Example

1. A Shape class can be the base class for several shapes including Circle, Rectangle, etc. Abstract classes make this possible even though it does not make sense to provide a specific implementation for an area, circumference or draw methods.
2. A collection of objects of a Pet hierarchy of classes can be processed by calling the same method in each object. Each object will perform its version of the method.

Use of Abstract Classes

- It is illegal to create objects from abstract classes since these classes are not fully defined. If Shape is an abstract class, then the following code will produce a compiler error.

  ```java
  Shape shape = new Shape(...); // Error
  ```

- It is legal to declare references (variables) of an abstract class.

  Subclasses of abstract classes have is-a relationship with the superclass. Hence it should be possible to use a reference of the abstract superclass to refer to objects of any concrete subclass.

  If Rectangle is a concrete subclass of Shape, then the following code is legal.

  ```java
  Shape rectangle = new Rectangle(...);
  ```

  Here, a reference, rectangle, of a superclass type, Shape, is used to refer to an object of the subclass, Rectangle, which is constructed with the call, new Rectangle(...).