Abstract class

If a class contain any abstract method then the class is declared as abstract class. An abstract class is never instantiated. It is used to provide abstraction. Although it does not provide 100% abstraction because it can also have concrete method.

What is Abstraction?

Abstraction is the process of **abstraction** in **Java** is used to hide certain details and only show the essential features of the object. In other words, it deals with the outside view of an object (interface).

Syntax :

```
abstract class class_name { }
```

Abstract method

Method that is declared without any body within an abstract class are called **abstract method**. The method body will be defined by its subclass. Abstract method can never be final and static. Any class that extends an abstract class must implement all the abstract methods declared by the super class.

Syntax :

```
abstract return_type function_name ();    // No definition
```

Example of Abstract class

```
abstract class A
{
 abstract void callme();
}
class B extends A
{
 void callme()
```

```
{
 System.out.println("this is callme.");
}
public static void main(String[] args)
{
 B b = new B();
 b.callme();
}
}
```

Output :

this is callme.

Abstract class with concrete(normal) method.

Abstract classes can also have normal methods with definitions, along with abstract methods.

```
abstract class A
{
 abstract void callme();
 public void normal()
 {
  System.out.println("this is concrete method");
 }
}
class B extends A
{
 void callme()
 {
  System.out.println("this is callme.");
```

```
}
public static void main(String[] args)
{
 B b = new B();
 b.callme();
 b.normal();
}
}
```

Output :

```
this is callme.
this is concrete method.
```

Points to Remember

1. Abstract classes are not Interfaces. They are different, we will study this when we will study Interfaces.
2. An abstract class may or may not have an abstract method. But if any class has even a single abstract method, then it must be declared abstract.
3. Abstract classes can have Constructors, Member variables and Normal methods.
4. Abstract classes are never instantiated.
5. When you extend Abstract class with abstract method, you must define the abstract method in the child class, or make the child class abstract.

Abstraction using abstract class

Abstraction is an important feature of OOPS. It means hiding complexity. Abstract class is used to provide abstraction. Although it does not provide 100% abstraction because it can also have concrete method. Lets see how abstract class is used to provide abstraction.

```
abstract class Vehicle
{
  public abstract void engine();
}
public class Car extends Vehicle {


   public void engine()
   {
      System.out.println("Car engine");
      //car engine implementation
   }


   public static void main(String[] args)
   {
      Vehicle v = new Car();
      v.engine();
   }
}
```

Output :

```
Car engine
```

Here by casting instance of **Car** type to **Vehicle** reference, we are hiding the complexity of **Car** type under **Vechicle**. Now the **Vehicle** reference can be used to provide the implementation but it will hide the actual implementation process.

When to use Abstract Methods & Abstract Class?

Abstract methods are usually declared where two or more subclasses are expected to do a similar thing in different ways through different implementations. These subclasses extend the same Abstract class and provide different implementations for the abstract methods.

Abstract classes are used to define generic types of behaviors at the top of an object-oriented programming class hierarchy, and use its subclasses to provide implementation details of the abstract class.