# Java Abstract class and methods

- **Abstract Class**

A class which is declared using abstract keyword known as abstract class. An abstract class may or may not have abstract methods. We cannot create object of abstract class.

An abstract class must be declared with an abstract keyword.

It can have abstract and non-abstract methods.

It cannot be instantiated.

It is used for abstraction.

**Syntax:**

abstract class class_name { }

- **Abstract method**

Method that are 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.

**Syntax:**

abstract return_type function_name (); //No definition

- **Points to Remember about Abstract Class and Method**

1. 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.

2. Abstract classes can have Constructors, Member variables and Normal methods.

3. Abstract classes are never instantiated.

4. When you extend Abstract class with abstract method, you must define the abstract method in the child class, or make the child class abstract.

- **Example of Abstract class**

In this example, we created an abstract class A that contains a method callme() and Using class B, we are extending the abstract class.

```
abstract class A
{
 abstract void callme();
}
class B extends A
{
 void callme()
 {
  System.out.println("Calling...");
 }
 public static void main(String[] args)
 {
  B b = new B();
  b.callme();
 }
}
```

- **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 behaviours at the top of an object-oriented programming class hierarchy, and use its subclasses to provide implementation details of the abstract class.