**Abstract classes in Java**

Abstract is a java modifier applicable for classes and methods in java but not for Variables.

**Illustration:** Abstract class

abstract class Shape

{

int color;

// An abstract function

abstract void draw();

}

Following are some important observations about abstract classes in Java.

1. An **instance** of an abstract class cannot be created.
2. **Constructors** are allowed.
3. We can have an abstract class without any abstract method.
4. There **can be final method in abstract class but any abstract method in class (abstract class) cannot be declared as final  or in simper terms final method cannot be abstract** itself as it will yield error: “Illegal combination of modifiers: abstract and final”
5. We are not allowed to create object for any abstract class.
6. We can define static methods in an abstract class
7. We can use abstract keyword for declaring top level classes (Outer class) as well as inner classes as abstract
8. **If a class contain at least one abstract method then compulsory we should declare class as abstract**
9. If Child class is unable to provide implementation to all abstract methods of Parent class then we should declare that Child class as abstract so that the next level Child class should provide implementation to remaining abstract method

**Abstract class vs Interface**

* **Type of methods:** Interface can have only abstract methods. An abstract class can have abstract and non-abstract methods. From Java 8, it can have default and static methods also.
* **Final Variables:** Variables declared in a Java interface are by default final. An abstract class may contain non-final variables.
* **Type of variables:**Abstract class can have final, non-final, static and non-static variables. The interface has only static and final variables.
* **Implementation:** Abstract class can provide the implementation of the interface. Interface can’t provide the implementation of an abstract class.
* **Inheritance vs Abstraction:** A Java interface can be implemented using the keyword “implements” and an abstract class can be extended using the keyword “extends”.
* **Multiple implementations:** An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces.
* **Accessibility of Data Members:** Members of a Java interface are public by default. A Java abstract class can have class members like private, protected, etc.

// Abstract class

abstract class Animal {

// Abstract method (does not have a body)

public abstract void animalSound();

// Regular method

public void sleep() {

System.out.println("Zzz");

}

}

// Subclass (inherit from Animal)

class Dog extends Animal {

public void animalSound() {

// The body of animalSound() is provided here

System.out.println("The Dog says: bhao bhao ");

}

}

class Main {

public static void main(String[] args) {

Dog myDog = new Dog(); // Create a Dog object

myDog.animalSound();

myDog.sleep();

}

}