Abstraction

Abstraction is a process of hiding the implementation details and showing only functionality to the user. It shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.

- Abstraction lets you focus on what the object does instead of how it does it.
- Abstract class in Java
- A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).
- A class which is declared as abstract is known as an abstract class. It can have abstract and non abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

Points to Remember

- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- An abstract class cannot be instantiated, which means you are not allowed to create an object of it. It can have constructors and static methods also.
- It can have final methods which will force the subclass not to change the body of the method.

Example

Let's say we have a class Animal that has a method sound() and the subclasses of it like Dog, Lion, Horse, Cat etc. Since the animal sound differs from one animal to another, there is no point to implement this method in parent class. This is because every child class must override this method to give its own implementation details, like Lion class will say “Roar” in this method and Dog class will say “Woof”. So when we know that all the animal child classes will and should override this method, then there is no point in implementing this method in the parent class. Thus, making this method abstract would be the good choice as by making this method abstract we force all the sub classes to implement this method(otherwise you will get compilation error), also we need not to give any implementation to this method in parent class. Since the Animal class has an abstract method, you must need to declare this class abstract. Now each animal must have a sound, by making this method abstract we made it compulsory to the child class to give implementation details to this method. This way we ensure that every animal has a sound.

```
//abstract parent class
abstract class Animal{
public abstract void sound(); //abstract method
}
public class Dog extends Animal{
public void sound(){
System.out.println("Woof");
}
public static void main(String args[ ]){
Animal obj = new Dog();
obj.sound();
```

```
}
}
//End of the program
```

ABSTRACT CLASS VS CONCRETE CLASS

A class which is not abstract is referred as Concrete class. In the above example, Animal is an abstract class and Cat, Dog & Lion are concrete classes.

Key Points

- An abstract class has no use until unless it is extended by some other class.
- If you declare an abstract method in a class then you must declare the class abstract as well. you can't have abstract methods in a concrete class. It's vice versa is not always true: If a class is not having any abstract method then also it can be marked as abstract.
- It can have a non-abstract method (concrete) as well.

ABSTRACT METHOD IN JAVA

A method without body (no implementation) is known as abstract method. A method must always be declared in an abstract class, or in other words you can say that if a class has an abstract method, it should be declared abstract as well.

This is how an abstract method looks in java:
public abstract int myMethod(int n1, int n2); //As you see this method has no body.

Rules of Abstract Method

1. Abstract methods don't have body, they just have method signature as shown above.
2. If a class has an abstract method it should be declared abstract, the vice versa is not true, which means an
abstract class doesn't need to have an abstract method compulsory.
3. If a regular class extends an abstract class, then the class must have to implement all the abstract methods of abstract parent class or it has to be declared abstract as well.

```
//abstract class and abstract methods
abstract class Sum{
abstract int sum(int n1, int n2);
abstract int sum(int n1, int n2, int n3);
void disp(){
System.out.println("Method of class Sum");
}
}
class AbstractDemo extends Sum{
int sum(int n1, int n2){
return n1 + n2;
}
int sum (int n1, int n2, int n3){
return n1 + n2 + n3;
```

```
}
public static void main(String args[]){
Sum obj = new AbstractDemo();
System.out.println("The sum of two numbers : "+obj.sum(5, 6));
System.out.println("The sum of three numbers : "+obj.sum(5, 6, 7));
obj.disp();
}
}
//End of the program
```