

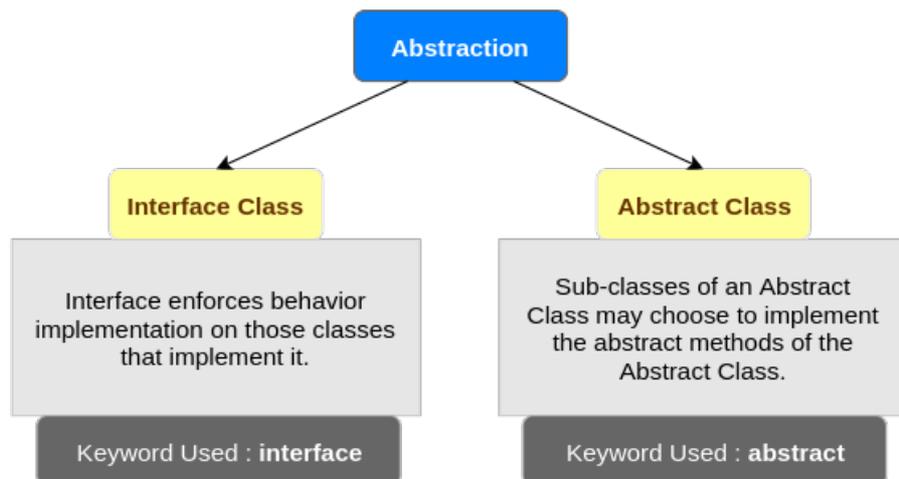
Abstraction in Java

Abstraction in Java

Abstraction in Java – Abstraction is one of the core concepts of Objected Oriented Programming. When there is a scenario that you are aware of what functionalities a [class](#) should have, but not aware of how the functionality is implemented or if the functionality could be implemented in several ways, it is advised to use abstraction. So, when some class is using(extending or implementing) this abstract class or methods, they may implement the abstract methods.

Abstraction in Java could be achieved with the following ways.

- [Interfaces](#)
- [Abstract Methods and Classes](#)



Interfaces in Java

Interfaces have similar structure as that of a class, but with abstract methods and variables that are static & [final](#) . Which means, the variables of interfaces are constants and methods do not have implementations. But an Interface in Java enforces definite implementations of methods/behaviors/routines in classes that implements the interface.

Following Car.java is an example of a Java Interface.

Car.java

```
public interface Car {
```

```
public void accelerate();
public void breaking();
}
```

Abstract Methods and Classes

An abstract method has only declaration part but no implementation or definition is provided.

An abstract class is a class that is declared abstract. And abstract class can have none or one or more abstract methods. But if there is at least one abstract method in a class, the class has to be declared abstract. And an important point to be noted is that an abstract class can never be instantiated, i.e., an object of abstract class type can never be created.

An abstract class can only be a super class for other normal classes or other abstract classes.

In the following program, Vehicle is an abstract class with abstract methods `accelerate()` and `breaking()`.

Vehicle.java

```
public abstract class Vehicle {
    public abstract void accelerate();
    public abstract void breaking();
}
```

Conclusion

In this [Java Tutorial](#), we have seen in brief the ways of implementing Abstraction in Java. In our next couple of tutorials, we shall learn about [Interfaces in Java](#) and [Abstract Methods and Classes in Java](#) in detail.

Java Tutorial

- ◆ [Java Tutorial](#)
- ◆ [Java Introduction](#)
- ◆ [Java Installation](#)
- ◆ [IDEs for Java Development](#)
- ◆ [Java HelloWorld Program](#)
- ◆ [Java Program Structure](#)
- ◆ [Java Datatypes](#)
- ◆ [Java Variable Types](#)

◆ Java Variable Types

◆ Java Access Modifiers

◆ Java Operators

◆ Java Decision Making

◆ Java Loops

◆ Java Array

◆ Java OOPs

◆ Java String

◆ Java Exception Handling

◆ Java File Operations

◆ Java Date & Time

◆ Java MySQL

◆ Java Random

◆ Java Math