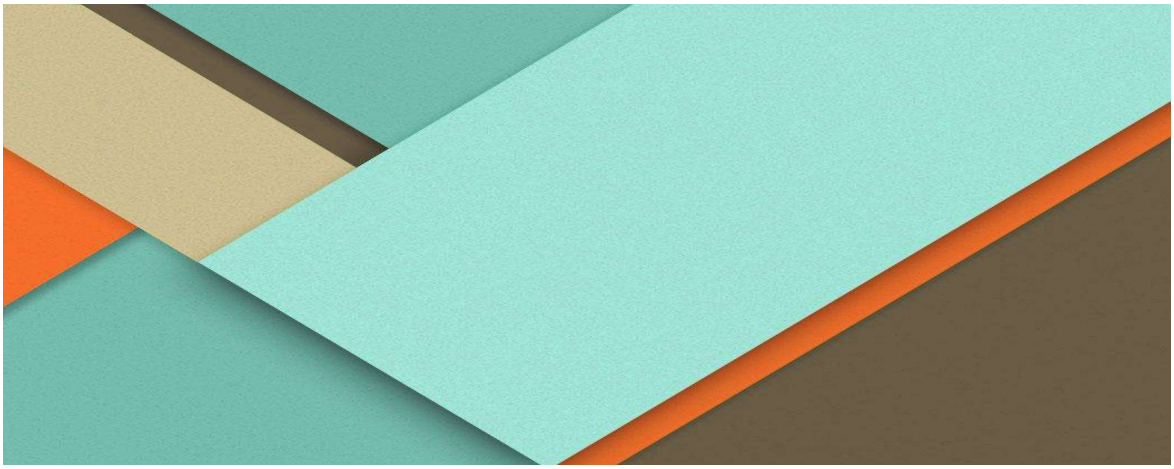
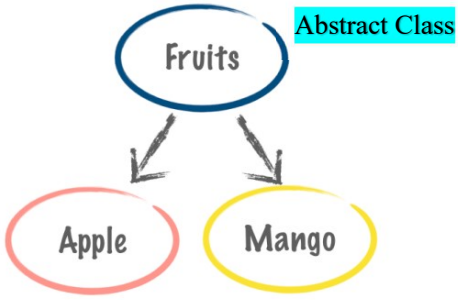


Abstract in JAVA



Abstract

Abstract is the modifier in JAVA which is applicable for methods and classes but not for variables.

Abstract method

- A method that is declared as abstract and does not have implementation (body) is known as abstract method. i. e. abstract method can have only method declaration but not implementation. Hence every abstract method declaration should compulsory ends with symbol ';' (semicolon).

- The responsibility of child class is provide implementation for abstract methods of parent class.

- When you declare abstract methods in parent class, it means we define guidelines to child classes which will describe the those methods who are to be compulsory to implement by child class.

Syntax of abstract method

```
abstract void display();
```

Only declaration required, no need to defined function body

Abstract class

- A class in JAVA that will be declared with a special keyword named "abstract" is known as abstract class.

- It can have abstract method (method without body) and non-abstract methods (method with body).

- It is essential that Abstract must be extended and its method must be implemented.

- It cannot be instantiated.

Syntax of abstract method

abstract class A{ }

Example 1: Simple JAVA program using abstract class

```
abstract class Parent
{
    public abstract void m1();
    public abstract void m2();
}

class TestChild extends Parent
{
    public void m1()
    {

    }
}
```

Output

```
Adarsh Shukla@AD-Desktop-Q3G3062 MINGW64 ~/Desktop
$ javac a.java
a.java:7: error: TestChild is not abstract and does not override abstract meth
od m2() in Parent
class TestChild extends Parent
^
1 error
```

We can handle this error either by:

Case 1: providing implementation for m2()

```
abstract class Parent
{
    public abstract void m1();
    public abstract void m2();
}

abstract class TestChild extends Parent
{
    public void m1()
    {
        // function defination
    }
}
```

Case 2: by declaring child class as abstract

```
abstract class Parent
{
    public abstract void m1();
    public abstract void m2();
}

class TestChild extends Parent
{
    public void m1()
    {
        // function defination
    }

    public void m2()
    {
        // function defination
    }
}
```

Example 2: Abstract class that has abstract method

In this example, BIKE is the abstract class that contains one abstract method RUN. Implementation of abstract method is provided by the HONDA class.

```
abstract class BIKE
{
    abstract void run();
    //abstract method

    void speed()
    {
        // non abstract method
    }
}

class HONDA extends BIKE
{
    void run()
    {
        System.out.println("Overridden Method run()");
    }
    public static void main(String args[])
    {
        BIKE obj = new HONDA();
        obj.run();
    }
}
```

Output

```
Adarsh Shukla@AD-Desktop-Q3G3062 MINGW64 ~/Desktop
$ javac Honda.java

Adarsh Shukla@AD-Desktop-Q3G3062 MINGW64 ~/Desktop
$ java HONDA
Overridden Method run()
```

Note: *If we use abstract methods, abstract class and interfaces in our program than it will cover in good programming practice.*

Abstraction in JAVA

Abstraction is a process in which user/programmer/developer hides the implementation data/details and shows only functionality or we can say required details to the user.
Another way, it shows only important things to the user and hides the internal details for example sending email, you just type the text and send the message. You don't know the internal processing about the message delivery.

Ways to achieve Abstraction

There are two ways to achieve abstraction in java

1. Abstract class (0 to 100%)
2. Interface (100%)