**Day 19**

**Abstraction**

* **Abstraction** is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only essential things to the user and hides the internal details.
* 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).
* Abstraction lets you focus on what the [object](https://www.javatpoint.com/object-and-class-in-java) does instead of how it does it.

There are two ways to achieve abstraction in java:

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

### **Abstract class in Java**

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.

#### Rules:

* An abstract class must be declared with an abstract keyword.
* It can have abstract and non-abstract methods.
* It cannot be instantiated.
* It can have [constructors](https://www.javatpoint.com/java-constructor) and static methods also.
* It can have final methods which will force the subclass not to change the body of the method.

Syntax

abstract class A

{

//body

}

### **Abstract Method in Java**

A method which is declared as abstract and does not have implementation is known as an abstract method.

Syntax

Abstract void show(); // no method body

Example

abstract class A

{

abstract void print();

}

class B extends A

{

void print()

{

System.out.println(“Abstract Method”);

}

public static void main(String args[])

{

B ob = new B();

ob.print();

}

}

Output

Abstract Method

**Constructors in Abstract Class**

abstract class A

{

A()

{

System.out.println(“Constructor”);

}

abstract void show();

void print()

{

System.out.println(“Print”);

}

}

class B extends A

{

void show()

{

System.out.println(“Show”);

}

}

class Test

{

public static void main(String args[])

{

A ob = new B(); // upcasting

ob.show();

ob.print();

}

}

Output

Constructor

Show

Print