**Abstraction in Java**

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, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery. Another example for abstraction is ATM. You have to only insert ATM and feed Pin number and give the withdrawal or deposit details. That is it , rest of all other things will be taken care by the ATM module.

**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. It should have atleast one abstract method

**Abstract Method in Java**

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

ABSTRACT METHOD in Java, is a method that has just the method definition but does not contain implementation. A method without a body is known as an Abstract Method. It must be declared in an abstract class. The abstract method will never be final because the abstract class must implement all the abstract methods.

**Syntax**

abstract class abstractclass\_name{

abstract void abstractmethod\_name();

}

**Example program**

abstract class Arith

{

protected int result;// To be inherited to derived class

abstract void calculate(int a, int b); // Abstract method.

void dispdata()// non-abstract method

{

System.out.println("This is to display data in abstract class");

}

}

class Addition extends Arith

{

// implementing abstract method

void calculate(int a, int b)

{

result = a + b;

}

// overriding the method dispdata() in the base class

void dispdata()// non-abstract method

{

System.out.println("Sum:" +result);

}

}

class Subtraction extends Arith

{

void calculate(int a, int b)

{

result = a - b;

}

void dispdata()// non-abstract method

{

System.out.println("Subtract:" +result);

}

}

class Multiplication extends Arith

{

void calculate(int a, int b)

{

result = a \* b;

}

void dispdata()// non-abstract method

{

System.out.println("Multiplication:" +result);

}

}

// Main Class

public class AbstractClass

{

public static void main(String[] args)

{

Addition a = new Addition();

Subtraction s = new Subtraction();

Multiplication m = new Multiplication();

a.calculate(20, 30);

a.dispdata();

s.calculate(10, 5);

s.dispdata();

m.calculate(10, 20);

m.dispdata();

}

}

o/p

Sum:50

Subtract:5

Multiplication:200