

# DAY-37

08 August 2023 15:38

## **ABSTRACT CLASS**

WHAT IS A CONCRETE METHOD ?

- A METHOD WHICH ARE HAVING IMPLMENTATION OR BODY IS KNOWN AS CONCRETE METHOD
- METHODS WHICH CONTAIN BODY WITH NO IMPLEMENATION IS KNOWN AS EMPTY IMPLEMENTATION METHOD.

```
class Test1
{
    public static void main(String[] args)
    {

    }
}
```

WHAT IS A CONCRETE CLASS ?

- A CLASS WHICH CONTAIN ONLY CONCRETE METHODS ARE KNOWN AS CONCRETE CLASS.

WHAT IS AN ABSTRACT ?

- WHEN THE USER WANTS TO DO SOME OPERATION BUT DOESN'T KNOW THE IMPLEMENTATION, SUCH SCENARIOS ARE CALLED AS ABSTRACT.

EXAMPLE :

```
public abstract void m1();
```

## **CASE-1**

```
package abstract1;
```

```
public abstract class Test1
{
    public abstract void m1();

    public static void m2()
    {
        System.out.println("inside the concrete method");
    }
}
```

```
package abstract1;
```

```
public class Test2 extends Test1
{
```

```

    public void m1()
    {
        System.out.println("inside the m1 method");
    }

    public static void main(String[] args)
    {
        Test2 a1 = new Test2();
        a1.m1();
        m2();
    }
}

```

## **CASE 2**

```
package abstract1;
```

```

public abstract class Test3
{
    public abstract void add();
    public abstract void diff();
}

```

```
package abstract1;
```

```

public abstract class Test4 extends Test3
{
    public void add()
    {
        int a = 1, b = 6;
        int c = a+b;
        System.out.println("The sum of a and b is "+ c);
    }
}

```

```
package abstract1;
```

```

public class Test5 extends Test4
{
    public void diff()
    {
        int a = 5, b = 1;
        int c = a-b;
        System.out.println("the difference is "+c);
    }

    public static void main(String[] args)
    {
        Test5 a1 = new Test5();
        a1.add();
        a1.diff();
    }
}

```

```
    }  
}
```

### **CASE - 3 : OVERRINDING OF METHOD IN ABSTRACT**

```
package abstract1;
```

```
public abstract class Chocolate  
{  
    public abstract void chocolateName();  
}
```

```
package abstract1;
```

```
public class DairyMilk extends Chocolate  
{  
    public void chocolateName()  
    {  
        System.out.println("Thank you for choosing Dairy Milk!!!!!!");  
    }  
}
```

```
package abstract1;
```

```
public class Kitkat extends Chocolate  
{  
    public void chocolateName()  
    {  
        System.out.println("Thank you for choosing KitKat!!!!!!");  
    }  
}
```

```
package abstract1;
```

```
public class Perk extends Chocolate  
{  
    public void chocolateName()  
    {  
        System.out.println("Thank you for choosing Perk!!!!!!");  
    }  
}
```

```
package abstract1;
```

```
import java.util.Scanner;
```

```
public class ChocolateShop  
{
```

```

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Hi welcome to chocolate shop");
    System.out.println("Enter the chocolate ID");

    int id = sc.nextInt();

    switch (id)
    {
        case 1:

            DairyMilk a1 = new DairyMilk();
            a1.chocolateName();
            break;

        case 2:
            Perk a2 = new Perk();
            a2.chocolateName();
            break;

        case 3:
            Kitkat a3 = new Kitkat();
            a3.chocolateName();
            break;

        default:
            System.out.println("Kindly enter valid id");
            break;
    }
}
}

```

