# POLYMORPHISM

(1) Object class:

- Object class is present in java.lang package, every class in java is directly or indirectly derived from the object class, if a class doesn't extend any other class then it is a direct child class of object and if extends another class then it is indirectly derived. Therefore the object class methods are available to all java classes, hence object class act as a root of the inheritance hierarchy in any java program.
- Object class methods:

  (a) getClass()

  (b) hashCode()

  (c) equals()

  (d) toString()

  (e) clone()

  (f) wait()

  (g) notify()

  (h) notifyAll()

  (i) finalize()

(2) polymorphism :

- we develop such features that depending on the situation it can take more than one form , called polymorphism because the word polymorphism means poly = many and morphism= many.
- There are two types of polymorphism:

  (a) Compile-time polymorphism:

  Compile-time polymorphism achieve by overloading.

Ex-

```
package org.example.opps_concept.polymorphism;

public class LearnPolymorphisom {
    public static void main(String[] args) {
        // polymorphisom
        Data d ;
        d= new ChildData();
        d.printData();

        //compile time polymorphism
        d.printData(5);
    }
}

class Data {

    String str = "sahil";
    public void printData(){
        System.out.println("print data...");
    }


    //compile time polymorphism
    public void printData(int times){
        for (int i=0; i<times; i++){
            System.out.println(str);
        }
    }
}

class ChildData extends Data {

    // polymorphisom
    @Override
    public void printData(){
        System.out.println("printing child data..");
    }
}
```

(b) Run-time polymorphism:

Run-time polymorphism achieve by overriding.

Ex-

```
package org.example.opps_concept.polymorphism;

public class LearnPolymorphisom {
    public static void main(String[] args) {
        // polymorphisom
        Data d ;
        d= new ChildData();
```

```
        d.printData();
    }
}

class Data {
    public void printData(){
        System.out.println("print data...");
    }
}

class ChildData extends Data {

    // polymorphisom
    @Override
    public void printData(){
        System.out.println("printing child data..");
    }
}
```