# Model paper of Java

# Q1- Explicate Polymorphism, Overloading & Overriding and also provide suitable coding examples.

**Ans:**

Polymorphism

* + Polymorphism means one name many forms.
  + One function behaves in different forms.
  + In other words, "Many forms of a single object is called Polymorphism."
  + For Example, A Human Being behaves like a SON in the house, at the same time that person behaves like an EMPLOYEE in office.

In Java polymorphism has two types:

* Compile-time Polymorphism(Overloading)
* Runtime Polymorphism(Overriding)

**1. Compile-time Polymorphism (Overloading):**

In is also Known as static polymorphism. This type of polymorphism is achieved by function overloading.

**Method Overloading**: When there are multiple functions with the same name but different parameters then these functions are said to be **overloaded.**

Functions may be overloaded by change in the number of arguments or/and a change in the type of arguments.

**// Java program for Method Overloading**

**// by Using Different Numbers of Arguments**

**// Class 1**

**class Helper {**

**// Method 1**

**// Multiplication of 2 numbers**

**public static int Multiply(int a, int b)**

**{ return a \* b;}**

**// Method 2**

**// Multiplication of 2 numbers**

Met/ // Multiplication of 3 numbers

**public static int Multiply(int a, int b, int c)**

**{ return a \* b \* c ;}**

**}**

**// Class 2 main class**

**class MAIN {**

**public static void main (String[] args) {**

**System.out.println (Helper.Multiply (2, 4));**

**System.out.println (Helper.Multiply (2, 7, 3));**

**}**

**}**

**2. Runtime Polymorphism (Overriding):**

This type of polymorphism is achieved by Method Overriding. Method

Overriding, on the other hand, occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be overridden.

**// Java Program for Method Overriding**

// Class 1

class Parent {

void Print(){

System.out.println("parent class");

}}

// Class 2

class subclass1 extends Parent {

// Class 3

void Print() { System.out.println("subclass1"); }}

class subclass2 extends Parent {

void Print(){

System.out.println("subclass2");}}

// Class 4

// Main class

class Main {

// Main driver method

public static void main(String[] args)

{ Parent a;

a = new subclass1();

a.Print();

a = new subclass2();

a.Print ();

}

}