Polymorphism in Java — Chapter#12

In Java OOP programming, Polymorphism means is we can perform a single action but in different ways. Polymorphism is derived from 2 Greek words: “poly” and “morphs”. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.

it occurs when we have many classes that are related to each other by inheritance. Like we specified in the previous chapter; Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to perform different tasks.

Why and when Importance of Java polymorphism

Implementation of polymorphism in Java provides the following benefits:

- The functionality of a method behaves differently in different scenarios.
- The behavior of a method depends on the data provided.
- It allows the same name for a member or method in a class with different types.
- Polymorphism supports implicit type conversion.

For example, think of a superclass called Animal that has a method called animalSound(). Subclasses of Animals could be Cats, Dogs and Birds - And they also have their own implementation of an animal sound (the dog bow Bow, and the cat meaion meaion, etc.)

```
class Animal
 {
   public void animalSound()
   {
    System.out.println("The animal
makes a Different sound");
  }
}
```

```
class Cat extends Animal
{
  public void animalSound()
  {
   System.out.println("The cat says:
meaion meaion");
  }
}
```

```
class Dog extends Animal
{
  public void animalSound()
  {
    System.out.println("The dog says:
bow Bow");
  }
}
```

```
class MainTest1PoliMorph
 {
   public staticvoid main(String[] args)
   {
   Animal myAnimal = new Animal();  // Create a Animal object
   Animal myCat = new Cat();  // Create a Cat object
   Animal myDog = new Dog();  // Create a Dog object
   myAnimal.animalSound();
   myCat.animalSound();
   myDog.animalSound();
  }
}
```

©Copy Right
http://www.sirmasood.com

Now we can create Cat and Dog objects and call the animalSound() method on both of them:

Example of Real-life: Polymorphism

A person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person possesses different behavior in different situations. This is called polymorphism.

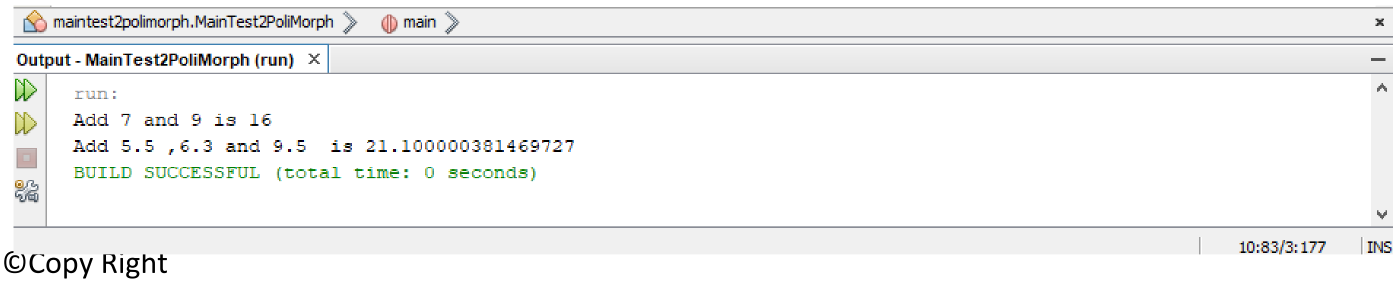
There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. We can perform polymorphism in java by method overloading and method overriding.

1) Compile-time polymorphism

It is also known as static polymorphism. This type of polymorphism is achieved by method/function overloading and operator overloading but ***operator overloading not supported in java***.

Method/Function Overloading: When there are multiple functions with the same name but different parameters then these functions are said to be **overloaded**. Functions can be overloaded by change in the number of arguments or/and a change in the type of arguments.

```
class Calculate {

   // Method with 2 integer parameters
   public int add(int a, int b)
   {

      // Returns addition of two integer numbers
      return a + b;
   }

   // Method 2
   // With same name but with 3 float parameters
   public float add(float a, float b, float c)
   {

      // Returns addition of float three numbers
      return a + b + c;
   }
}
// Main class
class MainTest2PoliMorph {

   public staticvoid main(String[] args)
   {
      Calculate calc = new Calculate();
      System.out.println("Add 7 and 9 is "+calc.add(7,9));
      System.out.println("Add 5.5 ,6.3 and 9.5  is "+calc.add(5.5f, 6.3f,9.3f));   }
}
```



maintest2polimorph.MainTest2PoliMorph > main >

Output - MainTest2PoliMorph (run)

```
run:
Add 7 and 9 is 16
Add 5.5 ,6.3 and 9.5  is 21.100000381469727
BUILD SUCCESSFUL (total time: 0 seconds)
```

10:83/3:177 INS

2) Runtime polymorphism

It is also known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at Runtime. 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.

Method overriding is when one of the methods in the super class is redefined in the sub-class. In this case, the signature of the method remains the same.

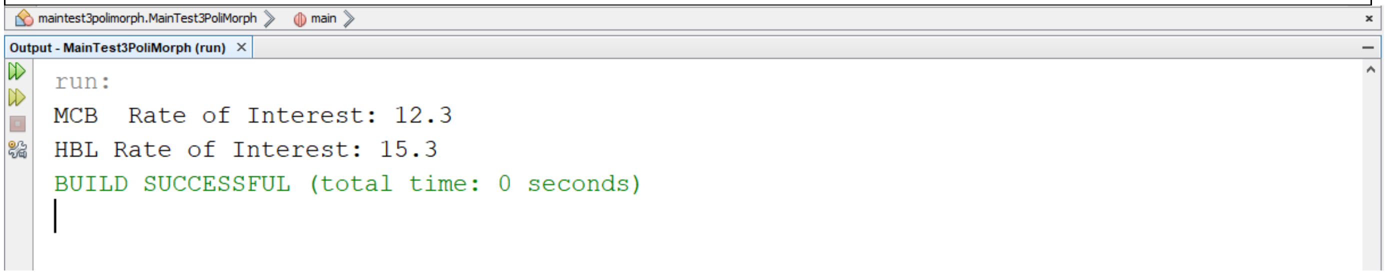
Consider a scenario where Bank is a class that provides a method to get the rate of interest. However, the rate of interest may differ according to banks. For example, MCB and HBL banks are providing 12.4% and 15.3% rate of interest.

```
class Bank
{
  public float getInterest()
  {
     return 0;
  }
}
```

```
class MCB extends Bank
{
  public float getInterest()
   {
      return 12.3f;
   }
}
```

```
Class HBL extends Bank
{
    public float getInterest()
     {
         return 15.3f;
      }
}
```

```
class MainTest3PoliMorph
{
     public static void main(String args[])
    {
       Bank b;
       b=new MCB();
           System.out.println("MCB  Rate of Interest: "+b.getInterest());
      b=new HBL();
          System.out.println("HBL Rate of Interest: "+b.getInterest());
    }
}
```


maintest3polimorph.MainTest3PoliMorph
main
Output - MainTest3PoliMorph (run)

```
run:
MCB  Rate of Interest: 12.3
HBL Rate of Interest: 15.3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exercise

Theory Questions

1. What is polymorphism and its types.
2. Why and when Importance of Java polymorphism
3. What is method/Function overloading.
4. Difference between the overloading and overriding.

Practical Questions

1. Write any example of polymorphism at run-time.

Objective MCQ's

1) the resolution of an _______ happens in the execution stage.
   a) overloading
   b) overriding
   c) both a and b
   d) nothing all

2) Method __________ is the process in which the class has two or more methods with the same name.
   a) Overloading
   b) Overriding
   c) Functions
   d) method

3) Polymorphism is derived from 2 Greek words: "_____" and "______".
   a) Pop and morphs
   b) Poly and method
   c) Poly and morph
   d) a and b both

4) Polymorphism supports ______ type conversion.
   a) extends
   b) public
   c) implicit
   d) explicit

5) Runtime polymorphism is also called _______ method dispatch.
   a) final
   b) private
   c) static
   d) Dynamic

6) operator overloading supported in java
   a) True
   b) False

©Copy Right