
LAB 6

Program to illustrate inheritance in Java

Definitions

Inheritance – It is one of the main features of OOPs. In inheritance, one class can “inherit” properties (member data and methods) from another class. The class that inherits the properties is the child class(sub-class). The class from which the child inherits properties is the parent class (superclass). In this way, our classes can reuse methods and data from existing classes.

extends – The keyword extends is used to denote when a class is a sub-class. The syntax is

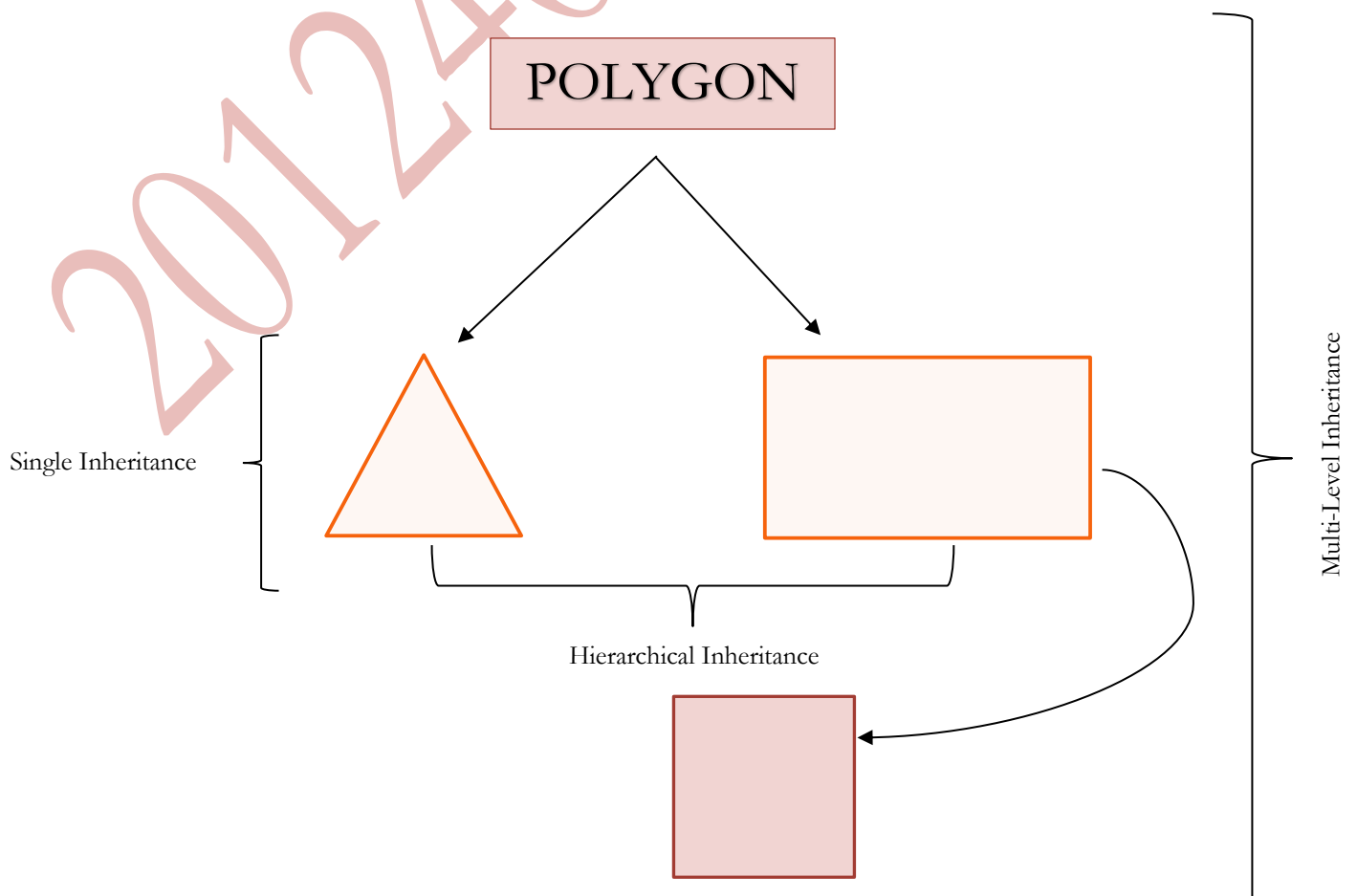
Syntax: `class sub-class_name extends parent-class_name {}`

super – The super keyword is used to call functions from the parent class.

Types of Inheritance – The types of inheritance in Java are

- Single inheritance – One child inherits from one parent
- Multi-level – One child inherits from one parent, and then acts as a parent for another sub class.
- Hierarchical – One parent class has multiple children

All these types of inheritance are shown in the diagram below.



Code

```
/* Program to illustrate
inheritance */

class Polygon {

    public Polygon() {
        System.out.println("I am a Polygon");
    }

    public void Sides() {
        System.out.println("I have n sides");
    }

    public void Area() {
        System.out.print("Area of polygon = ");
        System.out.println("(perimeter * apothem) / 2\n");
    }
}

class Triangle extends Polygon{
    public Triangle() {
        System.out.println("I am a triangle");
    }

    public void Sides() {
        super.Sides();
        System.out.println("n = 3");
    }

    public void Area() {
        System.out.println("Area of triangle = (base * height)/2\n");
    }
}

class Rectangle extends Polygon{
    public Rectangle() {
        System.out.println("I am a Rectangle");
    }

    public void Sides() {
        super.Sides();
        System.out.println("n = 4");
    }

    public void Area() {
        System.out.println("Area of rectangle = base * height\n");
    }
}
```

```

class Square extends Rectangle {
    public Square() {
        System.out.println("I am a Square");
    }

    public void Sides() {
        super.Sides();
        System.out.println("My sides are equal");
    }

    public void Area() {
        System.out.println("Area of a square = side * side\n");
    }
}

public class Lab6 {
    public static void main(String[] args) {

        System.out.println("Polygon :");
        Polygon P = new Polygon();
        P.Sides();
        P.Area();

        System.out.println("Rectangle :");
        Rectangle R = new Rectangle();
        System.out.println();
        R.Sides();
        R.Area();

        System.out.println("Square :");
        Square S = new Square();
        System.out.println();
        S.Sides();
        S.Area();
        System.out.println();

        System.out.println("Triangle :");
        Triangle T = new Triangle();
        System.out.println();
        T.Sides();
        T.Area();
    }
}

```

Explanation

- The code has been designed to show all types of inheritance, except multiple inheritance which is not a feature of java.
- The first class we created is Polygon. The constructor of this class prints “I am a Polygon”. It also has 2 methods, Sides and Area, which print information about the sides and the area of a polygon respectively.

- The next class is Triangle. When defining it, we write **class Triangle extends Polygon**. This means that Triangle is a child of polygon class. The constructor of Triangle prints “I am a triangle”. But, when we take a look at the output, we see it actually first prints “I am a Polygon” and then in the next line, it prints “I am a Triangle”. This is because, the constructor of the child class always first calls the parent class constructor, then proceeds to execute the code in its own constructor.
- In Triangle class, we again have the methods Area and Sides. The Area method simply prints the formula for area of a triangle. The Sides method first calls **super.Sides()**. This is a call to the Sides method in the Polygon class. This way, we get information on the sides of a polygon and a rectangle.
- The Rectangle class is created the same way as the Triangle class. They both inherit from Polygon. This is an example of Hierarchical inheritance.
- The Square class is next. It inherits from class Rectangle. This is an example of multi-level inheritance, because Rectangle inherits from Polygon and Square inherits from Rectangle.

Output

```
PS D:\NITJ\Sem 4\Java\Lab 6> javac Lab6.java
PS D:\NITJ\Sem 4\Java\Lab 6> java Lab6
Polygon :
I am a Polygon
I have n sides
Area of polygon = (perimeter * apothem) / 2

Rectangle :
I am a Polygon
I am a Rectangle

I have n sides
n = 4
Area of rectangle = base * height

Square :
I am a Polygon
I am a Rectangle
I am a Square

I have n sides
n = 4
My sides are equal
Area of a square = side * side

Triangle :
I am a Polygon
I am a triangle

I have n sides
n = 3
Area of triangle = (base * height)/2

PS D:\NITJ\Sem 4\Java\Lab 6> █
```