CSE 215: Programming Language II Lab

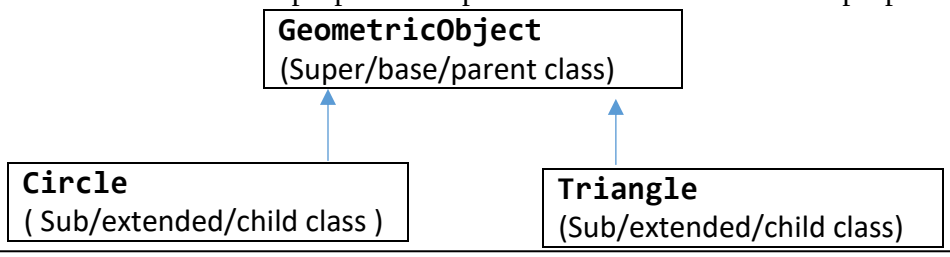
Lab – 9

Inheritance

Objective:

- To understand inheritance and its usage
- To utilize inheritance to ensure reusability of existing code

Inheritance has two purposes - reuse existing code, reduce code duplication. When common traits are found among two classes, define one as general/base/parent class and the other as specific/child class. Child class inherits the properties of parent class and adds its own properties.




```
package Lab11;
import java.util.Date;
public class GeometricObject {
  private String color = "White"; // setting white as default color
  private boolean filled;
  private Date dateCreated;
  public GeometricObject(){
    dateCreated = new Date();
  }
  public GeometricObject(String color, boolean filled) {
    this.color = color; // “this” refers to the current object
    this.filled = filled;
    dateCreated = new Date();
  }
  public String getColor() {
    return color;
  }
  public void setColor(String color) {
    this.color = color;
  }
  public boolean getFilled() {
    return filled;
  }
  public void setFilled(boolean filled) {
    this.filled = filled;
  }
  public Date getDateCreated() {
    return dateCreated;
  }
  public String toString(){
    return "Created on: "+dateCreated+" Color: "+color+" Filled: "+filled;
  }
}
```

The "extends" keyword:

public class Circle extends GeometricObject

The **"super" keyword**: The keyword super refers to the super class of the current class. It can be used for both constructors and methods.

Public Circle(String color, boolean filled, double radius) {
  super(color, filled);
  this.radius = radius;
}
	public String toString(){
	  return super.toString()+" radius: "+radius;
	}

Note: The compiler automatically puts a **super()** in subclass constructor to invoke the default constructor.

Lab Task

Design a class named **`Triangle`** that extends **`GeometricObject`**. The class contains:
 Three double data fields named **side1, side2, and side3** with default values 1.0 to denote three sides of the triangle.
 A no-arg constructor that creates a default triangle.
 A constructor that creates a triangle with the specified side1, side2, and side3.
 The accessor methods for all three data fields.
 A method named **`getArea()`** that returns the area of this triangle.
 A method named **`getPerimeter()`** that returns the perimeter of this triangle.
 A method named **`toString()`** that returns a string description with values of three sides of the triangle.

Write a test program that prompts the user to enter three sides of the triangle, a color, and a boolean value to indicate whether the triangle is filled. Create a Triangle object with these inputs. Display its area, perimeter, color, and true or false to indicate whether it is filled or not