Access Modifiers:

Private: Only Visible in the class that the member lives in\

Public: Accessible everywhere within the program

Using inheritance a class can inherit from one class, but also be the parent of another class. This means the information used for Employee can go to Analyst because every Analyst is an employee however, not every employee is an analyst. This makes the Analyst the subclass and the employee the superclass

class Shape {

private int numberOfSides;

public Shape(int numberOfSides) {

this.numberOfSides = numberOfSides;

}

public int getNumberOfSides() {

return numberOfSides;

}

public void setNumberOfSides(int numberOfSides) {

this.numberOfSides = numberOfSides;

}

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

super(0);

this.radius = radius;

}

public double getRadius() {

return radius;

}

}

class Rectangle extends Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

super(4);

this.length = length;

this.width = width;

}

public double getLength() {

return length;

}

public double getWidth() {

return width;

}

}

public class Main {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

System.out.println("A circle has " + circle.getNumberOfSides() + " sides.");

}

}