Lab Exercise 10

Chapter 10 Thinking in Objects
Chapter 11 Inheritance and Polymorphism

COMP217 Java Programming
Spring 2019
Instructor: Gil-Jin Jang

Text: Liang, Introduction to Java Programming, Tenth Edition
Chapters 10 and 11

Ex10-1
Submit TestWrongOverrding.java

Correct the LOGICAL errors

```
class X {
  X() { System.out.println(“default no-arg X”); }
  public String toString() { return new String(“X”); }
  public int comp(int n) { return n*2; }
  public int comp2(int n) { return n*4; }
}
class Y extends X {
  Y() { System.out.println(“constructor Y()”); }
  Y(int n) { this(); System.out.println(“constructor Y(“ + n + “)”); }
  public String tostring() { return new String(“Y”); }  // toString()?
  public void whoami() { System.out.println(toString()); }
  public void whoisparent() { System.out.println(super.toString()); }
  public int comp(int n) { return n*3; }
  public int comp2(double n) { return (int)(n*6); }
}

public class TestWrongOverriding {
  public static void main(String[] args) {
    Y y1 = new Y();   // (1)
    y1.whoami();      // (2)
    y1.whoisparent(); // (3)
    System.out.println(
     “comp(10) = “ + y1.comp(10) +       // (4)
     “\ncomp2(10) = “ + y1.comp2(10) +  // (5)
     “\ncomp2(10.0) = “ + y1.comp2(10.0));  // (6)
  }
}
```

```
$ java TestWrongOverriding
default no-arg X          (1)
constructor Y()           (1)
X                         (2)
X                         (3)
comp(10) = 30             (4)
comp2(10) = 40            (5)
comp2(10.0) = 60          (6)
```

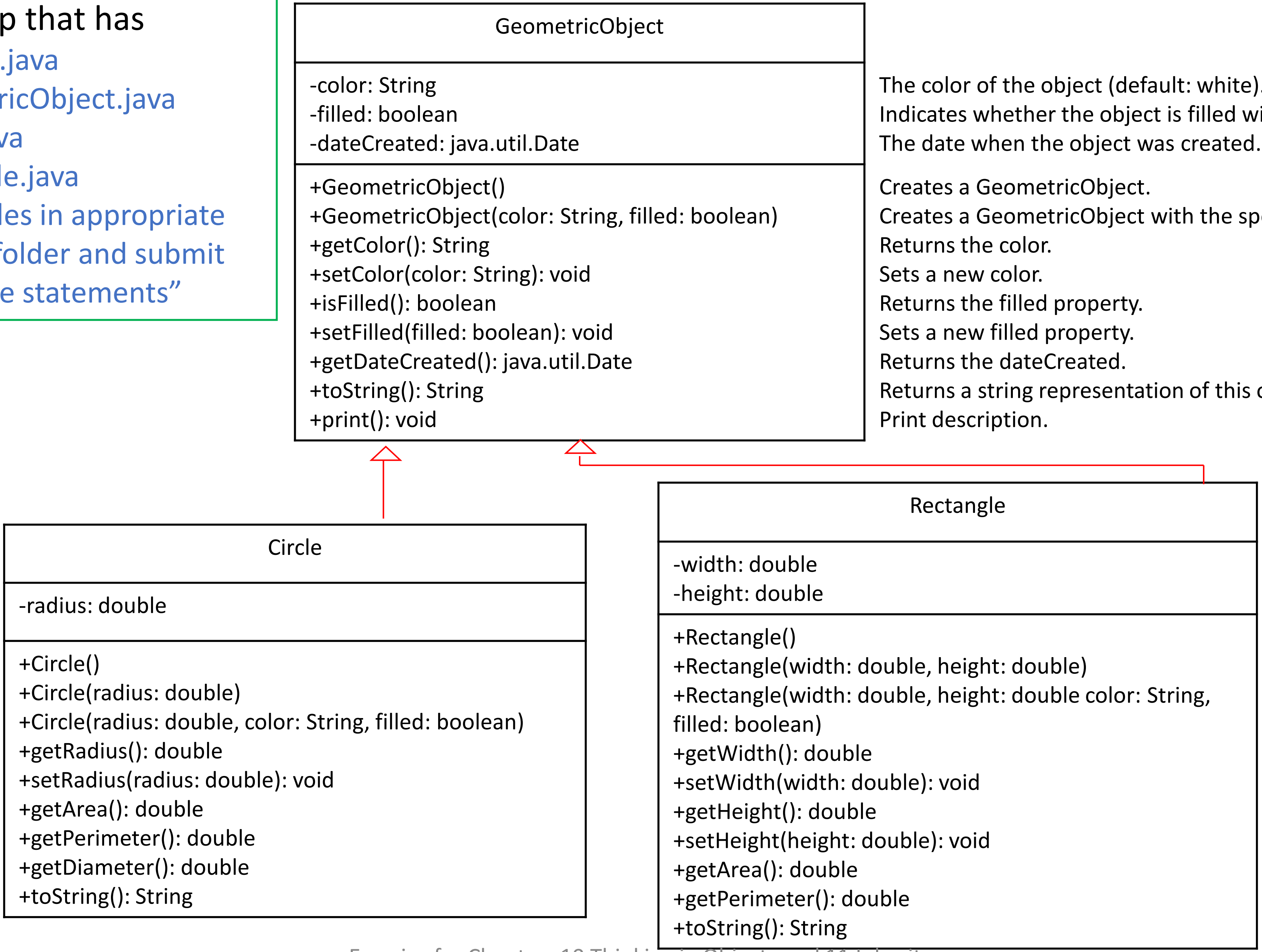
Ex10-2

Submit Ex10_2.zip that has

TestCircleRectangle.java
geometric/GeometricObject.java
geometric/Circle.java
geometric/Rectangle.java

- Place the java files in appropriate folders, zip the folder and submit
- Look at “package statements”

GeometricObject, Circle, Rectangle



GeometricObject	
-color: String	The color of the object (default: white).
-filled: boolean	Indicates whether the object is filled with a color (default: false).
-dateCreated: java.util.Date	The date when the object was created.
+GeometricObject()	Creates a GeometricObject.
+GeometricObject(color: String, filled: boolean)	Creates a GeometricObject with the specified color and filled.
+getColor(): String	Returns the color.
+setColor(color: String): void	Sets a new color.
+isFilled(): boolean	Returns the filled property.
+setFilled(filled: boolean): void	Sets a new filled property.
+getDateCreated(): java.util.Date	Returns the dateCreated.
+toString(): String	Returns a string representation of this object.
+print(): void	Print description.

Circle
-radius: double
+Circle() +Circle(radius: double) +Circle(radius: double, color: String, filled: boolean) +getRadius(): double +setRadius(radius: double): void +getArea(): double +getPerimeter(): double +getDiameter(): double +toString(): String

Rectangle
-width: double -height: double
+Rectangle() +Rectangle(width: double, height: double) +Rectangle(width: double, height: double color: String, filled: boolean) +getWidth(): double +setWidth(width: double): void +getHeight(): double +setHeight(height: double): void +getArea(): double +getPerimeter(): double +toString(): String

Ex10-2

```java
package geometric;
public class GeometricObject {
  private String color = "white";
  private boolean filled;
  private java.util.Date dateCreated;

  public GeometricObject() {
    dateCreated = new java.util.Date();
  }
  public GeometricObject(String color, boolean filled) {
    dateCreated = new java.util.Date();
    this.color = color;
    this.filled = filled;
  }
  public String getColor() { return color; }
  public void setColor(String color) {
    this.color = color;
  }
  public boolean isFilled() { return filled; }
  public void setFilled(boolean filled)
  { this.filled=filled; }
  public java.util.Date getDateCreated()
  { return dateCreated; }

  public String toString() {
    return "created on " + dateCreated
      + "\ncolor: " + color + " and filled: " + filled;
  }
  public void print() {
    System.out.println(toString());
  }
}
```

```java
package geometric;
public class Circle extends GeometricObject {
  private double radius;

  public Circle () { }
  public Circle (double radius)
  { this.radius = radius; }
  public Circle (
    double radius, String color, boolean filled) {
    this.radius = radius;
    setColor(color);
    setFilled(filled);
  }
  public double getRadius() { return radius; }
  public void setRadius(double radius)
  { this.radius = radius; }
  public double getArea() {
    return radius * radius * Math.PI;
  }
  public double getDiameter()
  { return 2 * radius; }
  public double getPerimeter()
  { return 2 * radius * Math.PI; }

  public String toString() {
    return "circle, radius " + radius + "\n"
      + super.toString();
  }
}
```

Ex10-2

```
package geometric;
public class Rectangle extends GeometricObject {
  private double width;
  private double height;

  public Rectangle () { }
  public Rectangle(double width, double height) {
    this.width = width;
    this.height = height;
  }
  public Rectangle (double width, double height,
      String color, boolean filled) {
    this.width = width; this.height = height;
    setColor(color); setFilled(filled);
  }
  public double getWidth() { return width; }
  public void setWidth(double width)
  { this.width = width; }

  public double getHeight() { return height; }
  public void setHeight(double height)
  { this.height = height; }

  public double getArea()
  { return width * height; }
  public double getPerimeter()
  { return 2 * (width + height); }
  public String toString() {
    return "rectangle, width " + width
      + " height " + height + "\n"
      + super.toString();
  }
}
```

```
import Circle;
import Rectangle;
public class TestCircleRectangle {
  public static void main(String[] args) {
    Circle circle = new Circle (1);
    System.out.println("A circle:" );
    System.out.println("The color is " + circle.getColor());
    System.out.println("The radius is “ + circle.getRadius());
    System.out.println("The area is “ + circle.getArea());
    System.out.println("The diameter is “+ circle.getDiameter());
    circle.print();

    Rectangle rectangle = new Rectangle (2, 4);
    System.out.println("\nA rectangle:“);
    System.out.println("The area is “ + rectangle.getArea());
    System.out.println("The perimeter is " + rectangle.getPerimeter());
    rectangle.print();
  }
}
```

```
$ java TestCircleRectangle
A circle:
...
circle, radius 1.0
created on Sat May 07 11:41:57 KST 2016
color: white and filled: false

A rectangle:
...
rectangle, width 2.0 height 4.0
created on Sat May 07 11:41:57 KST 2016
color: white and filled: false
```

Ex10-3
Submit CastingDemo2.java

GetArea methods for different shapes

Design the following shapes classes inherited from the GeometricObject

- Trapezoid (2 widths and height)
- Diamond (horizontal/vertical widths)
- Ellipse (a and b)

Define your own classes, and write your own getArea() methods

```
public class CastingDemo2 {
  public static void main(
      String[] args) {
    display(new Circle(3.5));
    display(new Rectangle(4,5));
    display(new Trapezoid(4,5,6));
    display(new Diamond(4,5));
    display(new Ellipse(4,5));
  }

  public static void display(
      Object obj) {
    /* FILL
       Write your own display method
       that displays areas of
       the shapes
    */
  }
}
```

Ex10-4
Test_toString.java

Modify the following code so that the output matches to the given

```
class Car {
  int speed;
  int gear;
  public String color;

  public Car() { /* FILL */ }
  // default values: 100 5 silver

  public Car(int, int, String)
  { /* FILL */ }
  // set the field values

  public String toString()
  { /* FILL */ }
}

class NamedCar extends Car {
  public String name;
  public NamedCar(String name)
  { super(); this.name = name; }
  public NamedCar(int, int,
    String color, String name)
  { /* FILL */ }

  public String toString() {
    return /* FILL */
    // use Car’s toString
  }
}
```

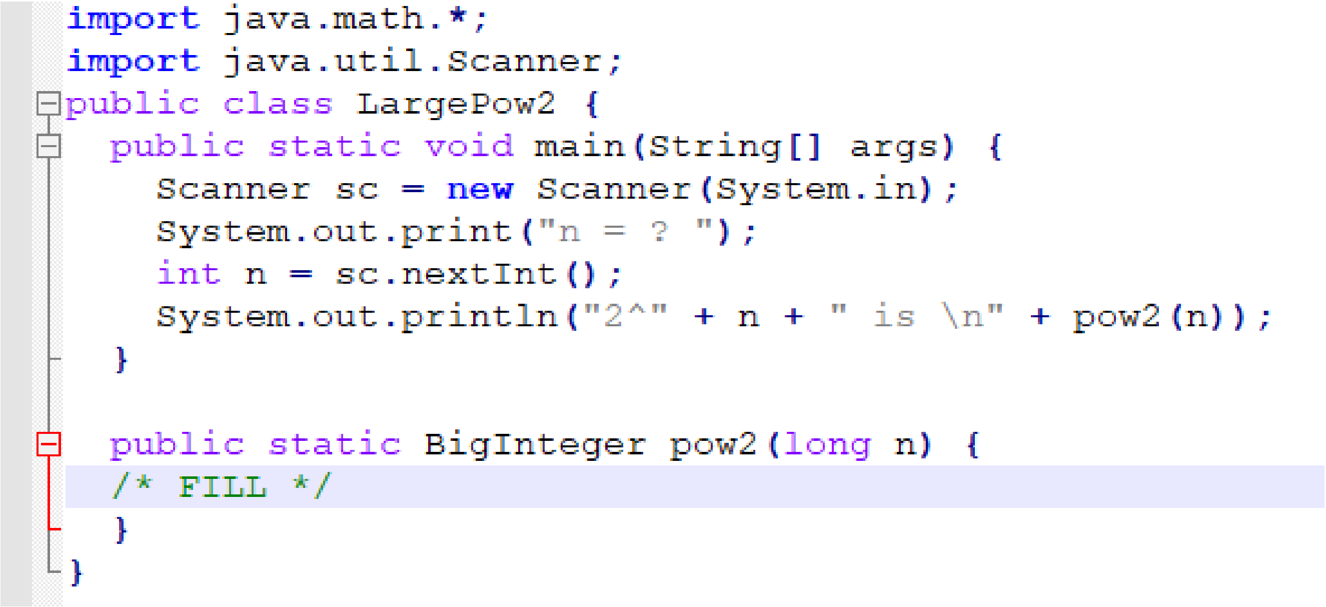
```
public class Test_toString {
  public static void main(String[] args) {
    Car c = new Car();
    System.out.println(c.toString());

    NamedCar c2 = new NamedCar(160, 8,
      "green", "Pony");
    System.out.println(c2.toString());
  }
}
```

```
/* expected results */
$ javac Test_toString.java
$ java Test_toString
Car: 100km/h 5gears silver
Car: 160km/h 8gears green Pony
```

Ex10-5
LargePow2.java

LargePow2.java



```
import java.math.*;
import java.util.Scanner;
public class LargePow2 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("n = ? ");
    int n = sc.nextInt();
    System.out.println("2^" + n + " is \n" + pow2(n));
  }

  public static BigInteger pow2(long n) {
  /* FILL */
  }
}
```

```
$ java LargePow2
n = ? 50
2^50 is
1125899906842624
$ java LargePow2
n = ? 100
2^100 is
1267650600228229401496703205376
$ java LargePow2
n = ? 500
2^500 is
32733906078961418700131896968275991522166
42046043064789483291368096133796404674554
88327009232590415715088668412756007100921
7256545885393053328527589376
$ java LargePow2
n = ? 1000
2^1000 is
10715086071862673209484250490600018105614
04811705533607443750388370351051124936122
49319837881569585812759467291755314682518
71452856923140435984577574698574803934567
77482423098542107460506237114187795418215
30464749835819412673987675591655439460770
62914571196477686542167660429831652624386
837205668069376
```

Ex10-6
WriteData.java
ReadData.java

WriteData.java

```
public class WriteData {
  public static void main(String[] args) throws Exception {
    java.io.File file = new java.io.File("scores.txt");
    if (file.exists()) {
      System.out.println("File already exists");
      System.exit(0);
    }

    // Create a file
    java.io.PrintWriter output = new java.io.PrintWriter(file);

    // Write formatted output to the file
    output.print("John T Smith ");
    output.println(90);
    output.print("Eric K Jones ");
    output.println(85);

    // Close the file
    output.close();
  }
}
```

Ex10-6
WriteData.java
ReadData.java

ReadData.java

```
import java.util.Scanner;

public class ReadData {
  public static void main(String[] args) throws Exception {
    // Create a File instance
    java.io.File file = new java.io.File("scores.txt");

    // Create a Scanner for the file
    Scanner input = new Scanner(file);

    // Read data from a file
    while (input.hasNext()) {
      String firstName = input.next();
      String mi = input.next();
      String lastName = input.next();
      int score = input.nextInt();
      System.out.println(firstName + " " + mi + " " + lastName + " " + score);
    }

    // Close the file
    input.close();
  }
}
```