

Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, clearly identify to which exam question it is related. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (10 points) True/False (T or F)

- _____ If `import java.util.Math` is not present before the class declaration, all occurrences of `Math` must be replaced with `java.util.Math`.
- _____ `Math.PI` is declared as a `static` attribute.
- _____ If a method is overloaded, it means that there are two methods with exactly the same name that accept a different set of arguments.
- _____ `JOptionPane.showMessageDialog(null, "Hi");` is a call to an object method.
- _____ Local variables declared in a constructor are only accessible within that constructor.
- _____ A `private` attribute is accessible to all methods defined within the class.
- _____ An accessor method must be declared as `private`.
- _____ A constructor is called only when an object is created.
- _____ The source code for all classes in the same package must be located in the same subdirectory/folder.
- _____ A class may have two attributes with exactly the same name as long as they are from different types and one of them is declared with `this.` in front of it. E.g.,

```
private double number;  
private int this.number;
```

2. (10 points) Clearly and succinctly describe how an object differs from a primitive.



3. (10 points) Clearly and succinctly distinguish between a value-returning method and a non-value-returning method. Give an example of each.

4. (10 points) Clearly and succinctly describe the role of the keyword `this`.

5. (10 points) Clearly and succinctly explain the benefits of object oriented design.

6. (15 points) Suppose a `Rectangle` class exists with two attributes: `height` (a double) and `width` (a double). Draw the memory diagram (like the diagrams drawn in lecture) that illustrate the state of memory at four points in the program on the right. Note: the zero argument constructor creates a unit square (both height and width are 1.0).

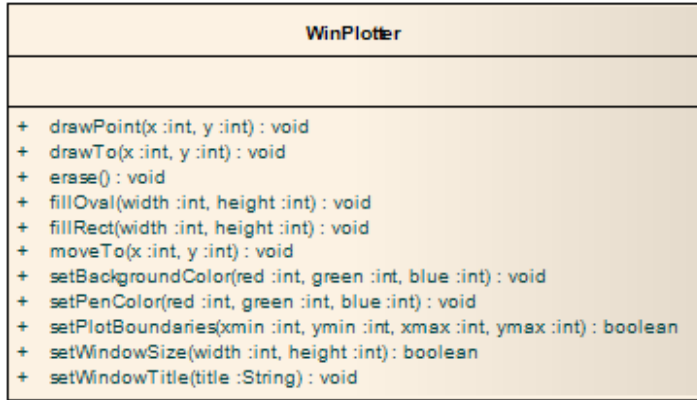
(a)

```
Rectangle r1 = new Rectangle ();  
Rectangle r2 = null;  
// (a)  
r2 = r1;  
r1.setWidth(50.2);  
// (b)  
int h = 3;  
int w = 5;  
r1 = new Rectangle(w, h);  
// (c)
```

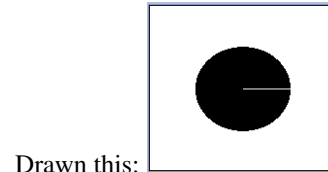
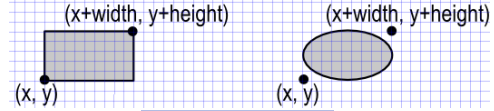
(b)

(c)

7. (15 points) Consider the following UML Class Diagram for the `WinPlotter` class (that you have not used before):



Assuming that the current location is (x, y) , the dimensions for the rectangle and oval drawn by the `fillRect()` and `fillOval()` methods are as follows:



Complete the following code segment that creates a `WinPlotter` object and uses it to draw a black circle with a radius of 50 in the middle of the drawing area. In addition, draw a white line from the center of the circle to the right-most part of the circle. Note:

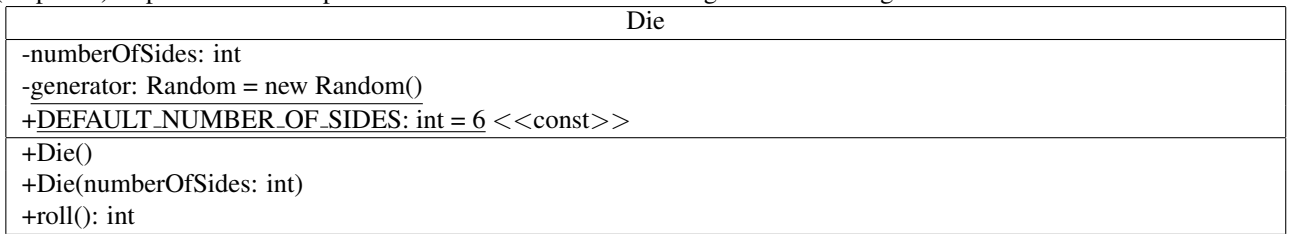
- The default background color is white
- The default pen color is black
- You can get black by setting red, green, and blue equal to 0
- You can get white by setting red, green, and blue to 255.
- The fill color for the rectangles and ovals is the same as the pen color

```

WinPlotter plotter = new WinPlotter();
int height = 200;
int width = 200;
plotter.setWindowSize(width, height);
plotter.setPlotBoundaries(0, 0, width, height);

```

8. (20 points) Implement the complete `Die` class show in the following UML class diagram.



The zero argument constructor should set the number of sides to the default number of sides. You may assume that `java.util.Random` has been imported.



Additional space — identify which problem your work is associated with.