

[May use one 8.5 × 11 inch sheet of paper for notes.] 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. (15 points) True/False (T or F)

- _____ If `import java.util.Scanner` is not present before the class declaration, all occurrences of `Scanner` must be replaced with `java.util.Scanner`.
- _____ If a method is overloaded, it means that there are two methods with exactly the same name that return different types (e.g., one returns `String` while the other returns `double`).
- _____ Local variables declared in a constructor are only accessible within that constructor.
- _____ It is illegal to place `this` on the left side of the assignment operator. E.g.,
`this = that;`
- _____ A constructor may call another constructor.
- _____ 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;`
- _____ A `private` attribute is accessible to all methods defined within the class.

2. (10 points) In the following code, circle all of the identifiers that represent primitives or classes, place a box around all identifiers that represent objects.

```
public String crazyStuff(Double a, Scanner scanner) {  
    JOptionPane.showMessageDialog(null, Math.sqrt(a));  
    return JOptionPane.showInputDialog(scanner.next());  
}
```

3. (10 points) Identify the error in the following class and explain why it is an error:

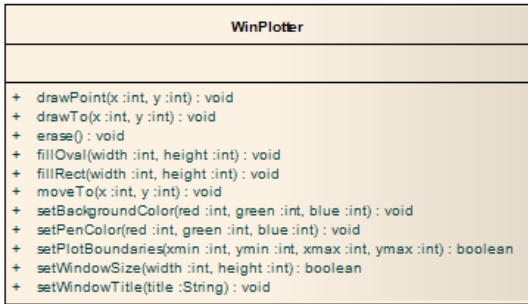
```
public class Exam2 {  
  
    private int questionNumber = 3;  
  
    public static String stringIt() {  
        return "" + questionNumber;  
    }  
}
```



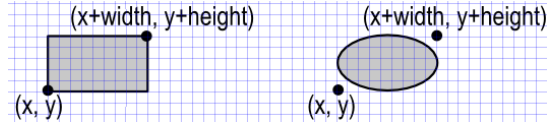
4. (20 points) In lecture we implemented a `Complex` class that had two object attributes: `real` and `imag`. Implement the `add` method for the `Complex` class that is used in the code below:

```
Complex a = new Complex(1.0, 1.0);
Complex b = new Complex(-2.0, -2.0);
Complex c = a.add(b);
System.out.println("(" + a + ") + (" + b + ") = (" + c + ")");
// Displays:
// (1.0 + 1.0i) + (-2.0 - 2.0i) = (-1.0 - 1.0i)
```

5. (20 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 above methods are as follows:



Using the names and parameters of the methods to infer meaning, complete the following code segment that creates a `WinPlotter` object and uses it to draw a 100×100 square in the middle of the drawing area. In addition, draw a line connecting the upper-left corner to the lower-right corner of the square.

```

WinPlotter plotter = new WinPlotter();
plotter.setWindowSize(200, 200);
plotter.moveTo(

```

6. (25 points) Complete the program below so that the following is true:

- If the user does not enter anything or enters a `String` with fewer than four characters, the program displays an error message and repeatedly displays the input dialog box.
- If the user selects cancel (then `null` is returned from the call to `showInputDialog()`), then the program terminates immediately.
- If the user enters a `String` with more than four characters, your program should display the square root of the total number of vowels (a, e, i, o, or u) entered by the user.
- The output must be displayed to the console.
- The number displayed should have three places after the decimal.
- Your program must not crash.

For example, if the user enters **"His cow is ready to calf too."**, your program should display: **"The square root of the number of vowels entered is 3.000"**.

```
public static void main(String [] args) {
    String input = null;
    boolean isDone = false;

    do {

        input = JOptionPane.showInputDialog("Enter at least four characters");

        if(input==null) {
            isDone = true;
        } else if(input.length()>4) {
            int count = 0;
            for(int i=0; i<input.length(); i+=1) {
                char c = input.charAt(i);
                if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u') {
                    count += 1;
                }
            }
            System.out.printf("The square root of the number of vowels entered is %.3f", Math.sqrt(count));
            isDone = true;
        } else {
            System.out.println("Error: must enter at least four characters");
        }
    } while(!isDone);
}
```



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



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