[**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. (10 points) True/False (**T** or **F**)

____ A variable that is declared within a method declaration is a *local variable*.

____ Local variables declared in a constructor are accessible from all other methods in the class.

____ Local variables declared in a constructor only are accessible from other constructors in the class.

____ In order to add a method to a class that has the same name as an already existing method, the number of arguments passed to the new method must be different than the number of arguments passed to the existing method.

____ A constructor may call another constructor.

____ A local variable with the same name as a class attribute will hide the attribute within the scope of the local variable.

____ A `private` attribute is accessible to all methods defined within the class.

2. (10 points) Explain the difference between an attribute that is declared `static` and one that is not.

3. (20 points) Complete the program below so that the following is true:

- If the user selects cancel (then `null` is returned from the call to showInputDialog()), then the program terminates immediately.
- If the user does not enter anything or enters a negative integer, the program displays and error messages a repeats the input prompt.
- If the user enters a positive integer, the program must display "Your age is: " followed by the age entered by the user. E.g., "Your age is: 18".
- Your program is allowed to crash if the user enters a non-integer value.

```
public static void main(String[] args) {
  int age = -1;
  String input = null;



  input = JOptionPane.showInputDialog("Enter your age");
```

4. (20 points) Recall the Complex class developed in lecture and sent to you via email. The class contained two `private` attributes: real and imag (both `double`s). Modify the toString() method such that it produces output consistent with the following examples:

- **3.2** when the real component is 3.2 and the imaginary component is 0.0.
- **3.2 + i5.7** when the real component is 3.2 and the imaginary component is 5.7.
- **3.2 - i5.7** when the real component is 3.2 and the imaginary component is -5.7.

5. (20 points) Consider the original implementation of the add() method from the Complex class:

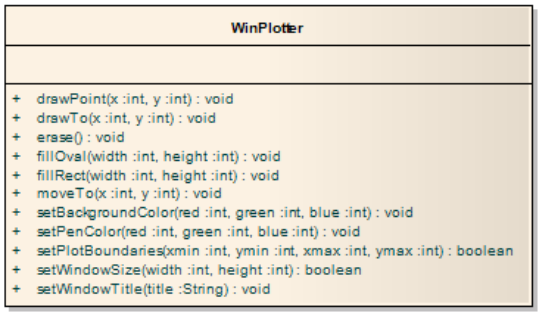
```
/**
 * Returns the result of adding two complex numbers.
 * @param arg The other complex number to be used in the addition
 * @return The sum of the complex numbers
 */
public Complex add(Complex arg){
  double rl = real + arg.real;
  double ig = imag + arg.imag;
  Complex result = new Complex();
  result.setReal(rl);
  result.setImaginary(ig);
  return result;
}
```

Rewrite the add() method so that the following code would produce "5 - i2.7" Note, nothing is returned by add(); instead, c1 after the call to add() is the sum of c1 and c2 before the call to add().

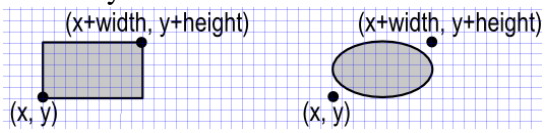
```
Complex c1 = new Complex(2, 7.3);
Complex c2 = new Complex(3, -10);
c1.add(c2);
System.out.println(c1.toString());
```

Note, nothing is returned by add(); instead, c1 = c1 + c2.

6. (20 points) Consider the following UML Class Diagram for the WinPlotter class:



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



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(
```

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