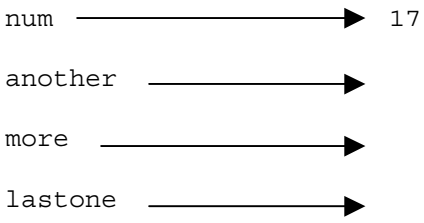
Answer all of the following questions. READ EACH QUESTION CAREFULLY. Answer each question in the space provided on these pages. Budget your time so you spend enough on the longer programming question at the end. There are a total of 50 points.

Keep your answers short and to the point. Good luck.

Question 1. (3 points) Complete the picture below to show the values that are bound to the following names after executing the following statements. The first one has been done for you.

```
int num = 17;
int another = num % 3 + 2;
int more = (num + 2) * 2;
int lastone = 2 * (num / 2);
```



Question 2. (8 points) For each of the following, indicate whether or not there are any errors in the code. If something is wrong, circle the error and indicate whether it is a syntax or semantic error.

(a) `int = 17;`

(b) `int n = 42;`

(c) `int a = "hello";`

(d) `String s = "howdy";`

(e) `String t = 17;`

(f) `String u = "17";`

(g) `double pi 3.14159;`

(h) `int bugnum = 100000+;`

Question 3. (5 points) In lecture we talked about what happens when a method call is executed. Here are the steps, but they are not in the correct order.

Indicate the correct order in which these steps occur by writing numbers from 1 to 5 in the blank spaces in front of each step.

_____ The method body's statements are evaluated until they are no more, or until a `return` statement is executed.

_____ The method's parameter names are bound to the corresponding values of the arguments.

_____ Control is passed back to the calling statement (the statement that called the method).

_____ Control is passed to the first statement in the method body.

_____ The arguments are evaluated left to right.

Question 4. (4 points) We've made the point several times that it is important that programs be well-written so that human readers can understand them.

(a) Why is this important? (Give at least one specific reason)

(b) Give two specific examples of things that a program author should do to help make code easier to understand. (Think about things that make it easier or harder for someone to understand how a program works and what it is doing.)

Question 5. (7 points) Here is a simple class with three methods.

```
public class Questionable {

   public void doThis( ) {

      int n = 42;                                  ______

      doSomethingElse(n);                          ______

      int m = n + 1;                               ______
   }

   public void doThat( ) {
      String fini = "almost done";                 ______
   }

   public void doSomethingElse(int n) {

      int one = 1;                                 ______

      doThat( );                                   ______

      int zero = one - one;                        ______
   }
}
```

Now, suppose that we create an instance (object) of class `Questionable` and call method `doThis()` for that object:

```
Questionable query = new Questionable();
query.doThis();
```

The question is, in what order do the statements in these methods ***finish*** execution when `doThis` is called? Fill in the blanks to the right of the statements to show the order in which the statements finish (1, 2, 3, …). If a method is not executed when `doThis` is called, leave the spaces for that method blank.

Question 6. (3 points) Normally, a name can only be created once. For example, the following code would generate a compiler error because the name x is declared twice:

```
Public class Tiny {

   public void zip() {
      int x = 12;    // fine
      …
      int x = 15;    // error – name already exists
   }
}
```

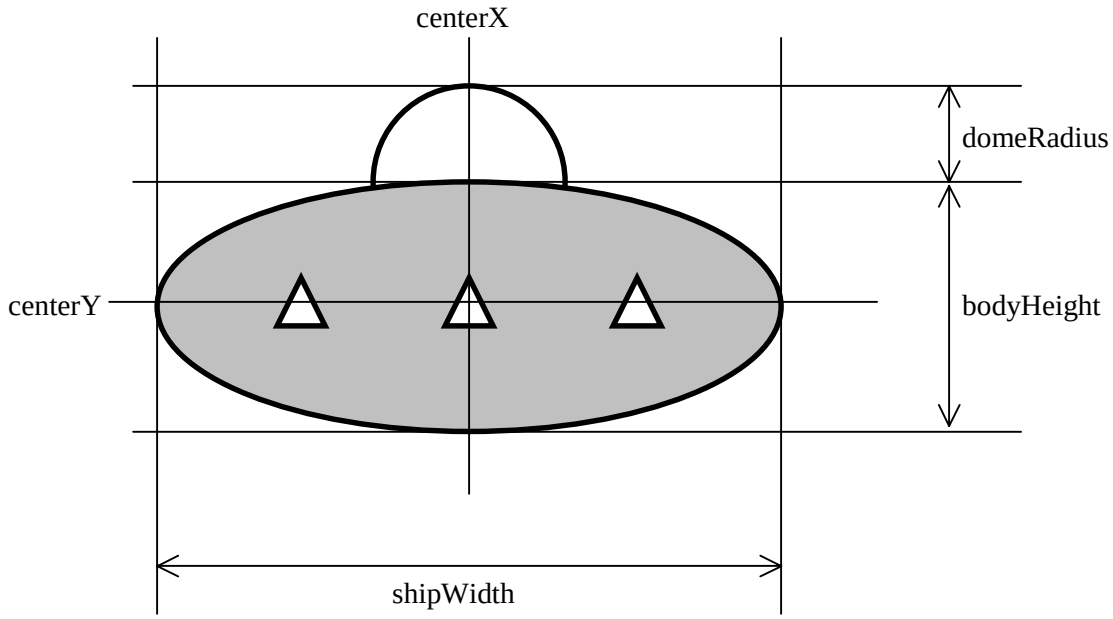
But it's possible for two different methods to use the same local name. For example,

```
public class Tim {

   public void one () {
      int n = 12;
      …
   }

   public void two() {
      int n = 142;
      …
   }
}
```

How is this possible? Why doesn't the second definition of `n` generate a "name already exists" error?

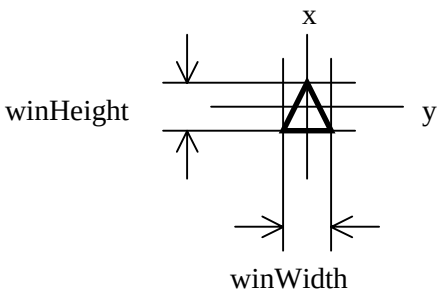
Question 7. (20 points) This longer question asks you to implement methods to draw a spaceship (UFO). Please read it completely before you start

Here's a picture of the UFO, with various dimensions and locations labeled.



The dome is an unfilled circle centered at the top center of the ship body. The location of the ship is specified by the coordinates of its center, which is also the center of the middle window.

The UFO has three triangular windows. They are centered vertically at the same y coordinate as the center of the ship's body. Horizontally, the windows are centered 1/4, 1/2, and 3/4 of the distance from the left edge of the ship to the right. Each window is a triangle, whose location and size are specified by its center coordinates, height, and width as follows:



Question 7 (cont). Write two methods that together draw the UFO. The first method should draw a single window (part a); then the second one (part b) should draw the complete UFO, using the method from part (a) as needed to draw the windows. The last page of the exam contains reference information for your use. You may want to tear it off so you don't need to do so much page flipping.

You should assume that a `GWindow` named `theWindow` is defined in an outer scope and you should use this window to draw your UFO (`theWindow.add(…)`, etc.). *Write the complete methods (specification and bodies) but don't worry about any of the surrounding class code.* Be sure to include appropriate heading comments.

(a) (6 points) Write a method named `drawUFOWindow` to draw one of the triangular windows. This method should have 4 integer parameters: the x and y coordinates for the center of the UFO window, and the width and height of the triangle.

(b) (14 points) Write a method named `drawUFO` to draw the complete UFO. This method should have 7 integer parameters: The x and y coordinates of the center of the ship's body; the height and width of the ship body, the radius of the dome that sits on top of the ship, and the height and width of the triangular windows. (Continue on the back side of the page if necessary.) For full credit, you **must** call method `drawUFOWindow` from part (a) to draw the three windows. The ship can be any solid color you like.

Reference information for question 7. Feel free to tear this page off so you don't have to flip back and forth while you're working on the question.

Parameter lists for new graphics objects:

```
Line(x1,y1,x2,y2)
Line(x1,y1,x2,y2,color)
Oval()
Oval(x,y,width,height)
Oval(x,y,width,height,color,fill)
Rectangle()
Rectangle(x,y,width,height)
Rectangle(x,y,width,height,color,fill)
Triangle()
Triangle(x1,y1,x2,y2,x3,y3)
Triangle(x1,y1,x2,y2,x3,y3,color,fill)
```

Standard Java colors:

```
Color.black, Color.blue, Color.cyan, Color.darkGray, Color.gray,
Color.green, Color.lightGray, Color.magenta, Color.orange,
Color.pink, Color.red, Color.white, Color.yellow
```

Copies of the UFO diagrams: