RIT | Golisano College of Computing and Information Sciences
School of Information

Name: ________________________________

ISTE-120
Lab 04: Implementing Classes

Exercise 1 - A Class with Class Attributes (4 points)

The exercise must be completed during the lab period.

Download the "Lab04StarterFiles.zip" file from myCourses and extract it to get Student.java.

1. Add to the `Student` class a `toString` method so that the code:
   ```
   Student stu = new Student("Jane", "Brown", 182765, 2.333);
   System.out.println(stu.toString());
   ```
   will print:
   ```
   Brown, Jane
   ID: 182765  GPA: 2.3
   ```

   NOTE: The GPA is formatted via `String.format` using an appropriate format as in Lab03. There should be one digit to the left and one digit to the right of the decimal point.

2. Write another class, `Address.java` with attributes:
   ```
   private String street;
   private String city;
   private String state;
   private int zip;
   ```

   This represents an address. You should provide a constructor that expects a parameter for each of the above attributes and initializes the object to those values.

3. Also, write a `toString()` method that will print out the address. If the address is:
   ```
   Address anAddress = new ("13 Flower St.", "Pultneyville",
      "NY", 14386);
   ```
   then
   ```
   System.out.println(anAddress.toString());
   ```
   will print:
   ```
   13 Flower St.
   Pultneyville, NY  14386
   ```

4. Write a third class, `StudentRecord`, that has two attributes:
   ```
   Student stu;
   Address addr;
   ```
   and two constructors. The first constructor is given a `Student` object and an `Address` object to initialize the attributes. The second constructor is given a first name, a last name, a student ID, a gpa, a street address, a city, a state, and a zipcode and uses these to initialize the attributes.

5. Provide in `StudentRecord` a `toString()` method that will return the `toString()` of the `Student` followed by a new line and then the `toString()` of the `Address`.

6. Finally, write a main program in the file `TestStudentRecord.java` that will initialize and print out the information for two students:

   ```
   Bluestone, Barbara
   ID: 23686 GPA: 2.8
   Main St.
   Any Town, NY  14539

   Broderick, Matthew
   ID: 39872 GPA: 3.5
   34 Worsted Pl
   NoPlaceVille, UT  29873
   ```

   For `Barbara Bluestone`, create a `Student` object, then an `Address` object and pass these to the appropriate constructor to create the `StudentRecord`.

   For `Matthew Broderick`, pass all of the information components to the other constructor and let that constructor create the `Student` and `Address` objects.

   Print both `StudentRecord` objects out using the `toString()` method of the `StudentRecord` class.

   All input information may be hard coded into your program.

Signature: ______________________________ Date: _____________
Have your instructor or TA sign here when Exercise 1 works correctly.

Exercise 2 – Simple Animation (6 points)

<u>If you do not complete this exercise during the lab period, you need to complete the work outside of the lab period and bring the completed work to the lab next week.</u>

In this exercise, we are going to draw an arrow, pointing up, and animate it - it will move from the bottom of the window to the top.

1. First, we need a class to represent the arrow. The arrow will be a composite class - that is, it contains two or more classes within it. In our case, besides the canvas on which it is drawn, the arrow will have two visible pieces - a shaft (a rectangle) and a head (a triangle). To draw the arrow will take two steps: draw the shaft then draw the head. It will be similar for erasing it (erase the head, then the shaft).

   Write the Arrow class …

   The Arrow class will have the attributes:
   ```
   private Canvas canvas;        // the canvas on which to draw
   private Triangle head;        // a filled, B&W head
   private Rectangle shaft;      // an unfilled, B&W shaft
   ```

   Draw the arrow as an unfilled rectangle (the shaft) with a filled triangle (the head) centered on top of it. The arrow should appear centered (left to right) and at the bottom of the canvas. The shaft should be 10x100 pixels and the arrow head should be in a rectangle of 50x50 pixels.

2. The constructor for the Arrow will accept one parameter - the Canvas on which to draw. It will instantiate the head and the shaft so that they will appear at the bottom center of the Canvas. The constructor will <u>not</u> draw either component at this time.

3. Write a main class, called Controller. Have the main program call the Controller constructor:

   ```
   public class Controller {
      // attributes declared HERE

      public static void main(String[] args) {
         new Controller();
      }

      public Controller() {
         // your problem is to write this code
      }
   }
   ```

Declare two attributes. One should be an object of class Canvas and the other an object of class Arrow. The constructor is where all the work will be done. Because it is not static, like the main program, it will be able to access the attributes.

Have the constructor create (with **new**) the two attributes. The Canvas should be 300x700 pixels and should be passed to the constructor for the Arrow class.

4. In the Controller class, declare that the main method, and the constructor, throw Exceptions:

```
      public static void main(String[] args) throws Exception {
```
AND
```
      public Controller() throws Exception {
```

This will keep errors that we will learn how to handle later from bothering us now.

5. Going back to the Arrow class, we need a number of methods:

```
      public void draw()
      public void erase()
      public void moveUp()
```

The `draw` method will draw the arrow on the Canvas. This means drawing both the shaft and the head (using the draw method from Canvas).

The `erase` method will erase the arrow on the Canvas. This means erasing both the shaft and the head (using the erase method from Canvas).

The `moveUp` method will move the position of both the shaft and the head up 50 pixels. Do this by using the `setYInt()` method of both the `shaft` and the `head` attributes to change the Y value of each. Use the `getYInt()` method of these attributes to get the current Y value and decrement it by 50.

6. Finally, in the Controller constructor, after Canvas and Arrow have been initialized, insert the code:

```
      arrow.draw();      // draw the arrow at the bottom of the screen
      Thread.currentThread().sleep(100);  // delay a little
```

Follow this with the code:

```
      arrow.erase();     // erase the arrow
      arrow.moveUp();    // move the arrow up 50 pixels
      arrow.draw();      // redraw the arrow
      Thread.currentThread().sleep(100);  // delay a little
```

Repeat these 4 lines 10 times. Compile and run this program.

7. Answer these questions:

a) What is the purpose of the erase, moveUp, draw groupings of operations?

answer:

These are methods used to manipulate the Triangle and Rectangle drawing without having to duplicate code.
The groupings will simulate that the arrow is traveling upward by erase its previous position,moving it up, and then drawing it at its new location

b) What is the purpose of the `Thread.currentThread().sleep(100)` lines? What happens if they are removed?

answer: This allows the human eye to see the change of the arrow as it travels up. The computer computes at a speed in which is extremely fast.
Having the computer pause and continue with the code, gives the smooth transitions of the arrow, and being able to see each location of the arrow.

Signature: ______________________________ Date: _____________
Have your instructor or TA sign here when Exercise 2 works correctly.