(a) List any penalties associated with lab assignments that were described on the general course policies page.

(b) Implement a method that will accept an ArrayList of Strings and return the shortest String in the ArrayList. In the event of a tie, you should return the String that is closest to the front of the ArrayList. For example, if the ArrayList contains the following:

```
"The", "quiz", "was", "easy", "to", "do"
```

the method should return **to**. If the ArrayList is `null` or empty, an empty String should be returned.

Recall the Shape class developed in lecture:

```
+-------------------------------------------------------+
|                          Shape                        |
+-------------------------------------------------------+
| -color: Color                                         |
| -centerLocation: Point2D                              |
+-------------------------------------------------------+
| +Shape(color: Color, centerX: double, centerY: double) |
| +getArea(): double                                    |
| +toString(): String                                   |
+-------------------------------------------------------+
```

Recall further that the getArea() method just returns zero and the toString() method returns something like this:

```
Color: 0x00ffffff
Location: Point2D [x = 0.0, y = 0.0]
```

Provide a complete implementation (excluding `package` or `import` statements) for a subclass called RightTriangle. The class should model a right triangle that oriented in any arbitrary direction (see examples drawn on whiteboard).

Suppose First is an interface with two methods: iMethodA() and iMethodB(). Suppose further that Parent is a class with the following methods: a constructor that accepts a String as an argument, an overridden equals() method, and pMethodA().

(a) Show the class declaration (just the first line) of a Child that inherits from Parent and implements the First interface.

(b) List the method signatures for all of the methods that **must** be implemented by the Child class.

(c) List all of the reference types that can refer to a Child object. I.e., what can replace XXX in the following line of code?

```
XXX obj = new Child();
```

(a) Sketch what will be displayed when a class that extends Application contains the following method implementation:

```
@Override
public void start(Stage stage) {
    Pane root = new VBox();
    root.getChildren().addAll(
        new Label("This seems like"),
        new Button("a"),
        new Button("useless"),
        new Label("graphical"),
        new Button("user"),
        new Label("interface."));
    stage.setTitle("Quiz 4");
    stage.setScene(new Scene(root, 400, 400);
    stage.show();
}
```

(b) List two replacements for VBox in the first line of the start() method above and indicate how the UI would change for each replacement.

```
public static void main(String[] args) {
  Scanner in = new Scanner(System.in);
  try {
    int i = in.nextInt();
    double x = -1.0;
    System.out.println(i*x);
    x = in.nextInt();
    System.out.println(i*x);
  } catch(RuntimeException e) {
    System.out.println("ouch");
  } finally {
    System.out.println("done");
  }
}
```

Consider the code above and indicate what will be displayed if the user were to enter the following input.

(a)

```
3 5
```

(b)

```
thirty three
```

(c)

```
8 30.0
```

List the UI controls that your group has been investigating this week in lecture. Suppose you are interviewing for a summer internship. What would you say to convince your interviewer of what you learned about the UI controls this week.

Suppose the following method has already been implemented:

```
public static List<Complex> getSpecified(List<Complex> list, Predicate<Complex> predicate);
```

Complete the following code using a lambda expression for the predicate that will cause result to contain all of the elements in list that a negative real component. For example, if the list contains the following complex numbers: $3 + 2i$, $-1 + 0i$, $-8.2 - 2.1i$, and $0 - 9i$, then result should contain $-1 + 0i$ and $-8.2 - 2.1i$.

```
public static void quiz8(List<Complex> list) {


    List<Complex> result =









    System.out.println(result);
}
```

Consider the following interface:

```
@FunctionalInterface
public interface Transformable {
    Color newColor(Color color);
}
```

(a) Suppose Image image exists. Complete the call to transformImage() (see part (b)) using a lambda expression that will produce brighterImage. Each pixel in the resulting image should have the value returned by calling .brighter() on it.

```
Image result = transformImage(image,
```

(b) Implement the following method such that each pixel in the image returned is a transformed version of the corresponding pixel in the imaged passed to the method. The transformation is defined by the Transformable passed to the method.

```
public static Image transformImage(Image image, Transformable transform) {
```