Show all of your work clearly in the space provided. Be sure to **read each problem carefully**. Note that the exam is double sided. In this exam, wk5.ArrayList refers to the ArrayList implementation created in lecture.

1. (15 points) True/False (**T** or **F**)

____ JavaFX provides classes that can be nested within one another to create complex graphical user interfaces.

____ Any of the following can have a fx:id attribute in an FXML file: Label, Button, and TextField.

____ Any of the following can have an onAction attribute in an FXML file: Label, Button, and TextField.

____ The Alert class can be used to generate a pop-up window.

____ One can convert from a File to a Path by casting. E.g., Path path = (Path) file.

____ It is not possible to store primitive values in a binary file.

____ Because the wk5.ArrayList is an indexed collection, you can access its elements using a subscript (the square brackets operator).

____ The FXMLLoader instantiates an instance of the controller class and all of the GUI components specified in the FXML file.

____ The Application class is an abstract class.

____ Primitive types cannot be used as a generic type.

____ A $O(2^n)$ algorithm always takes longer to complete than a $O(n)$ algorithm when run on the same hardware.

____ A lambda expression can be used to create an instance of a functional interface.

____ Functional programming facilitates declarative programming techniques that specify what we want done but not how.

____ When analyzing the $O()$ time complexity of an algorithm, we seek to characterize the worse-case performance of the algorithm as the input size grows beyond the amount of computer memory available.

____ The add(int index, E element) method for wk5.ArrayList class is $O(n)$.

2. (10 points) Without using any loops implement the following method that, given a list of strings, will return a list of integers containing the length of each string. For example,

```
["I", "hope", "this", "is", "correct"]
```

should become

```
[1, 4, 4, 2, 7]
```

You may assume that the list and all strings in the list are not `null`.

```
public static List<Integer> questionTwo(List<String> nums) {















}
```

3. (10 points) Consider the following interface:

```
@FunctionalInterface
interface Examinator {
  double apply(double[] numbers);
}
```

Write a lambda expression that implements the interface above in such a way that the first value in the array of numbers is always returned.

```
  Examinator alwaysFirst =
```

4. (10 points) Sketch what the UI specified in the following FXML file:

```
<HBox fx:controller="msoe.Controller" prefHeight="200" prefWidth="200"
        xmlns="http://javafx.com/..." xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <VBox>
      <children>
        <TextField fx:id="addend1" />
        <Label text="+" />
        <TextField fx:id="addend2" />
      <children>
    </VBox>
    <Button onAction="#handleButton" text="Calculate" />
    <Label fx:id="result" />
  </children>
</HBox>
```

5. (10 points) Write code that will create a new TextField called field and assign an handler to it that will replace the contents of the text field with lowercase letters whenever enter is pressed in the text field. Note: this is not an FXML question.

6. **(a)** (15 points) Recall that our wk5.ArrayList had one private attribute: data. Implement the add(int index, E element) method. Be sure to throw any exceptions as appropriate. You may not make use of any other methods in the class.

(b) (5 points) State the time complexity for the method implemented in **(a)**. Justify your answer.

7. (10 points) Consider the following method:

```
public static void populateList(List<Integer> list) {
  list.add(null); // 1
  list.add(3);    // 2
  list.add(1, 1); // 3
}
```

Draw a memory diagram (similar to the ones drawn in lecture) showing the contents of the list after each line of the method above is run when passed an empty wk5.ArrayList.

After the first call to add().

After the second call to add().

After the third call to add().

8. (5 points) Show how to get the second command line argument within the JavaFX start() method. (Explain in as much detail as possible if you don't remember sepecific names of attributes/methods.)

9. (5 points) Explain why we cannot store primitive types in our wk5.ArrayList.

5. (5 points) Explain why, as a general rule, exceptions should be caught in the controller.