

[Student may use one single-sided 8.5 × 11 inch sheet with handwritten reference material.]

Show all of your work clearly in the space provided or on the additional page at the end of the exam. Be sure to **read each problem carefully**. Note that the exam is double sided. Due to time constraints, you are not required to document your source code.

1. (20 points) Circle TRUE or FALSE

- TRUE | FALSE The `EventHandler` interface uses a generic to specify the type of event that is being handled, e.g., `ActionEvent`.
- TRUE | FALSE The `EventHandler` interface declares only one method: `handle()` that accepts one argument.
- TRUE | FALSE In a JavaFX program, the class containing `main()` must extend the `Application` class.
- TRUE | FALSE In a JavaFX program, the class containing `main()` must override the `start()` method.
- TRUE | FALSE If an FXML file has the same filename (but different extension) as the class with the `main()` method, the graphical user interface described in the FXML file is automatically loaded into the primary stage.
- TRUE | FALSE A `FlowPane` is used to graphically represent the flow of data through an output stream.
- TRUE | FALSE FXML represents a declarative style of programming graphical user interfaces.
- TRUE | FALSE It is possible to throw an exception from within the body of any `if` statement.
- TRUE | FALSE If an exception is thrown outside of a `try` block, all execution in the method is terminated and the method that called this method is responsible for handling the exception.
- TRUE | FALSE Once an exception is caught, the remainder of the method (starting after all of the catch blocks) is executed.

2. (5 points) Consider the following:

```
try {  
    // ...  
} catch (Class1 e) {  
    // ...  
} catch (Class2 e) {  
    // ...  
}
```

What constraints, if any, are there on the relationship between `Class1` and `Class2`.

3. (5 points) Describe how checked exceptions differ from unchecked exceptions.

4. (15 points) Consider the following code:

```
int[] nums = {1, 2, 3};
String in = JOptionPane.showInputDialog(null, "Enter an integer");
System.out.println("TY");
int index = Integer.parseInt(in);
System.out.println("Thank you");
try {
    System.out.println(nums[index]);
} catch (NullPointerException e) {
    System.out.println("null's make me sad");
} catch (NumberFormatException e) {
    System.out.println("Hint: integers are stuff");
} finally {
    System.out.println("Finally");
}
System.out.println("thank U");
```

Determine the output if the user:

(a) enters 1 and clicks “OK”

(b) enters one and clicks “OK”

(c) enters -1 and clicks “OK”

5. Suppose a `Complex` class exists and has two instance variables: `double real` and `double imag`.

(a) (10 points) Implement the `Complex.read(DataInputStream in)` method that reads two `doubles` from a binary input stream, creates a `Complex` object, and returns it.

(b) (10 points) Suppose that `Complex.read(Scanner in)`, which reads a `Complex` number from a text file, has already been implemented. Complete the program below that asks the user to enter a filename and then reads one complex number (stored as text) from the file and displays it to the console.

If any file-related exception is thrown, the program should display “File problems” and then terminate.

```
public static void main(String[] ignored) {  
    String filename = JOptionPane.showInputDialog(null, "Enter a filename");
```

6. (10 points) Sketch what the GUI described in the following FXML file looks like:

```
<VBox xmlns:fx="http://javafx.com/fxml/1" fx:controller="Exam2Controller">
  <children>
    <HBox>
      <children>
        <Button onAction="#handler1" text="Press" />
        <Button fx:id="button" text="Button" />
      </children>
    </HBox>
    <Label fx:id="label" text="text me" />
    <Button onAction="#handler2" text="Click" />
    <Button onAction="#handler3" text="Mouse" />
  </children>
</VBox>
```

7. (20 points) Implement the controller class that can be used with the FXML file in the previous problem. The controller should enable the following behavior:

- When the button labelled **Press** is clicked, the text on the button labelled **Button** is changed to **button**.
- When the button labelled **Click** is clicked, the label is changed from **text me** to **not so bad**.
- When the button labelled **Mouse** is clicked, **yeah** is displayed on the console.



Additional work area for any problem. Clearly identify to which problem the work on this page is related.