

COE 593 – COE Application

Welcome to the Midterm Exam
Wednesday November 02, 2016

Instructor: Dr. Wissam F. Fawaz

Name: _____

Student ID: _____

Instructions:

1. This exam is **Closed Book**. Please do not forget to write your name and ID on the first page.
2. You have exactly **55 minutes** to complete the **4** required problems.
3. Read each problem carefully. If something appears ambiguous, please write your assumptions.
4. Points allocated to each problem are shown in square brackets.
5. Put your answers in the space provided only. No other spaces will be graded or even looked at.

Good Luck!!

Problem 1: Lambdas and Streams (10 minutes) [20 points]

- 1) Which of the following methods can be used to perform an operation on each key-value pair of a Map data structure?
 - a. `forAll`
 - b. `forNext`
 - c. **`forEach`**
 - d. None of the above

- 2) Which of the following static methods of the `Collectors` class returns a `Collector` that collects objects in a given classification into a `List`?
 - a. `list()`
 - b. `toList()`
 - c. `intoList()`
 - d. **None of the above**

- 3) Which of the following methods of the `Stream` class maps objects to double values returning a `DoubleStream`?
 - a. `doubleMap`
 - b. `toDouble`
 - c. **`mapToDouble`**
 - d. `toDoubleStream`

- 4) Which of the following statements is false?
 - a. **The `Files` class method `lines` creates a `String` containing the lines of text from a given file**
 - b. `Stream` method `flatMap` receives a `Function` that maps an object into a stream, for instance a line of text into words
 - c. `Pattern` method `splitAsStream` uses a regular expression to tokenize a `String`
 - d. `Collectors` method `groupingBy` with three arguments receives a classifier, a `Map` factory, and a downstream `Collector`

- 5) `Map` method `entrySet` returns a `Set` of `Map.Entry` objects containing the `Map`'s
 - a. values
 - b. keys
 - c. **Both of the above**
 - d. None of the above

- 6) Which of the following statements is false?
- To convert an `IntStream` into a `Stream<Integer>`, the method `toStream` of `IntStream` can be used**
 - `SecureRandom` method `ints` with three arguments creates a finite `InStream` of random `int` values
 - Function static method `identity` creates a `Function` that returns its argument
 - None of the above is false
- 7) Which of the following makes it possible to include a non-abstract method inside as part of an interface in Java 8?
- `Optional`
 - `@FunctionalInterface`
 - `default`**
 - None of the above
- 8) Which of the following methods of the `Comparator` class uses a key extractor to perform comparison according to some extracted key?
- `compare`
 - `compared`
 - `comparing`**
 - None of the above
- 9) Which of the following methods of the `Stream` class can be used to cut an infinite sequence of values to size?
- `filter`
 - `generate`
 - `mapToInt`
 - `limit`**
- 10) Consider an `Optional<String>` object called `lineAsOptional`. What does the following method call return if `lineAsOptional` holds a value of `null`?
`lineAsOptional.orElse(" ").length()`
- The method call causes a `NullPointerException` to be thrown
 - 1**
 - 0
 - None of the above

Problem 2: Miscellaneous (10 minutes) [20 points]

- 1) When developing a Java-FX application using Eclipse, three files are created, namely a Java-FXML file, a file containing the application's main class, and a file containing the application's
 - a. control class
 - b. model class
 - c. view class
 - d. **None of the above**

- 2) A Java-FX application's main class is directly derived from
 - a. Main
 - b. **Application**
 - c. Object
 - d. None of the above

- 3) Which of the following static methods of the `ByteBuffer` class can be used to create a `ByteBuffer` object?
 - a. `wrap`
 - b. `allocate`
 - c. **Both of the above**
 - d. None of the above

- 4) Which of the following statement is true about `Buffers` of the `java.nio` package?
 - a. **The `rewind` method always resets the position back to 0 while the limit remains intact**
 - b. The `flip` method switches a buffer from read mode into write mode
 - c. Both of the above are false
 - d. None of the above

- 5) Which of the following can be used to compute the Hamming distance separating two `int` variables called `a` and `b` respectively?
 - a. `a | b`
 - b. `a & b`
 - c. **`a ^ b`**
 - d. None of the above

- 6) Which of the following does not match the regular expression
"(?!)\p{P}"?
- a. ,
 - b. .
 - c. ;
 - d. **None of the above**
- 7) What output does the following code fragment produce?
- ```
String searchObject = "xxfooxxxxfoo";
Pattern pattern = Pattern.compile(".*+foo");
Matcher matcher = pattern.matcher(searchObject);
while(matcher.find())
 System.out.println(matcher.group());
```
- a. xxfoo  
xxxxfoo
  - b. xxfooxxxxfoo
  - c. **No output**
  - d. None of the above
- 8) Which of the following matches the regular expression:  
"a(?!b)"?
- a. cb
  - b. **a**
  - c. ab
  - d. None of the above
- 9) What does a Java-FX keyboard event handler receive when a keyboard key is pressed?
- a. ActionEvent
  - b. KeyPressedEvent
  - c. EventHandler<ActionEvent>
  - d. **None of the above**
- 10) Which of the following represents the middle mouse button in Java-FX?
- a. MouseEvent.PRIMARY
  - b. MouseEvent.SECONDARY
  - c. MouseEvent.MIDDLE
  - d. **None of the above**

**Problem 3: IO streams (20 minutes) [30 points]**

1. Design and implement a Java program that reads from the user an `int` value and then makes use of bitwise operators to produce an output message indicating whether or not the input value is a power of 2.

Specifically, your program must employ **four different strategies** that utilize bitwise operators to verify whether or not the input value is a power of 2.

```
public class BitwiseOperators {
 Scanner scan = new Scanner(System.in);
 int a;
 System.out.println("Enter an int:");
 a = scan.nextInt();

 if((a & (a-1)) == 0)
 System.out.println(a + " is a power of two");
 else
 System.out.println(a + " is not a power of two");

 if((a & (~a+1)) == a)
 System.out.println(a + " is a power of two");
 else
 System.out.println(a + " is not a power of two");

 int c = a;
 while(((c & 1) == 0) && c > 1)
 c >>= 1;
 if(c == 1)
 System.out.println(a + " is a power of two");
 else
 System.out.println(a + " is not a power of two");

 c = a;
 int onesCount = 0;

 while(c > 0 && onesCount <=1) {
 if((c & 1) == 1)
 onesCount++;
 c >>= 1;
 }

 if(onesCount == 1)
 System.out.println(a + " is a power of two");
 else
 System.out.println(a + " is not a power of two");
}
}
```

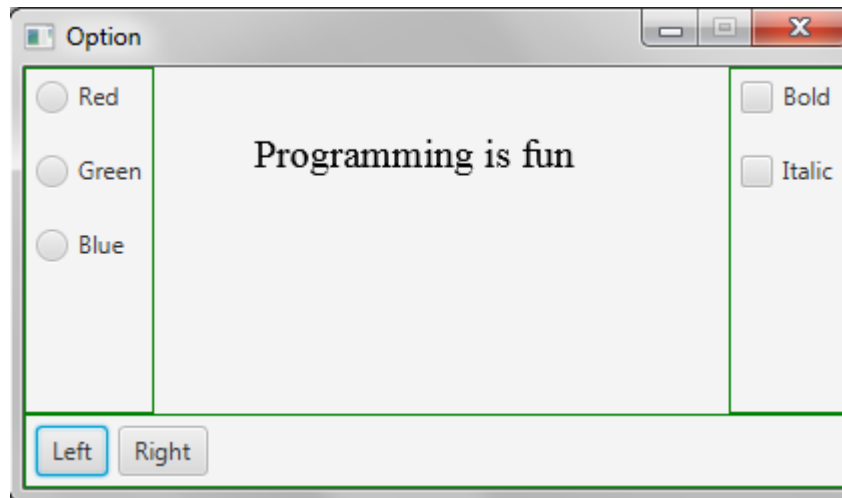
2. Design and implement a Java program that copies **half of** the content of a text file called "srcFile.txt" into a target file called "trgtFile1.txt" using the `FileChannel` and `ByteBuffer` classes of the `java.nio` package. Your implementation should leverage the file random access capability of the said classes to accomplish the desired task.

```
import java.nio.channels.FileChannel;
import java.nio.file.Path;
import java.nio.file.Paths;
import static java.nio.file.StandardOpenOption.*;

import java.io.IOException;
import java.nio.ByteBuffer;
public class CBasedFileManagement {
 public static void main(String[] args) throws Exception {
 Path srcPath = Paths.get("srcFile.txt");
 Path trgtPath = Paths.get("trgtFile1.txt");
 FileChannel srcChannel = FileChannel.open(srcPath);
 FileChannel trgtChannel = FileChannel.open(trgtPath,
 CREATE, WRITE);
 ByteBuffer buffer = ByteBuffer.allocate((int)
 srcChannel.size()/2);

 srcChannel.read(buffer);
 buffer.flip();
 trgtChannel.write(buffer);

 srcChannel.close();
 trgtChannel.close();
 }
}
```

**Problem 4: Java-FX (15 minutes) [30 points]**

Write a Java-FX application that displays the graphical user interface shown above to the end user. In particular, your application should allow the user to:

- Use the buttons to move the text that is displayed in the center either to the left or to the right,
- Use the two check boxes to specify whether the text should be displayed in bold or italic, and
- Use the radio buttons to choose the color of the message.



```

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
public class Main extends Application {
 private BorderPane mainPane;
 private HBox paneForButtons;
 private VBox paneForRadioButtons, paneForCheckBoxes;
 private Pane paneForText;
 private Button btLeft, btRight;
 private RadioButton rbRed, rbGreen, rbBlue;
 private CheckBox chkBold, chkItalic;
 private Text text = new Text(50, 50, "Programming is fun");
 private ToggleGroup colorToggleGroup;
 private void createTextPane() {
 paneForText = new Pane();
 paneForText.getChildren().add(text);
 }
 private void createPaneForButtons() {
 paneForButtons = new HBox(5);
 paneForButtons.setPadding(new Insets(5, 5, 5, 5));
 paneForButtons.setStyle("-fx-border-color: green");
 btLeft = new Button("Left");
 btRight = new Button("Right");
 btLeft.setOnAction(e -> text.setX(text.getX()-10));
 btRight.setOnAction(e -> text.setX(text.getX()+10));
 paneForButtons.getChildren().addAll(btLeft, btRight);}
 private void createPaneForCheckBoxes() {
 paneForCheckBoxes = new VBox(20);
 paneForCheckBoxes.setPadding(new Insets(5, 5, 5, 5));
 paneForCheckBoxes.setStyle("-fx-border-color: green");
 chkBold = new CheckBox("Bold");
 chkItalic = new CheckBox("Italic");
 Font fontNormal = Font.font("Times New Roman",
 FontWeight.NORMAL, FontPosture.REGULAR, 20);
 Font fontBold = Font.font("Times New Roman",
 FontWeight.BOLD, FontPosture.REGULAR, 20);

```

```

Font fontItalic = Font.font("Times New Roman",
FontWeight.NORMAL, FontPosture.ITALIC, 20);
Font fontBoldItalic = Font.font("Times New Roman",
FontWeight.BOLD, FontPosture.ITALIC, 20);
EventHandler<ActionEvent> handler = e-> {
 if(chkBold.isSelected() &&
 chkItalic.isSelected())
 text.setFont(fontBoldItalic);
 else if(chkBold.isSelected())
 text.setFont(fontBold);
 else if(chkItalic.isSelected())
 text.setFont(fontItalic);
 else
 text.setFont(fontNormal);
};
chkBold.setOnAction(handler);
chkItalic.setOnAction(handler);
paneForCheckBoxes.getChildren().addAll(
 chkBold, chkItalic);}
private void createPaneForRadioButtons() {
 paneForRadioButtons = new VBox(20);
 paneForRadioButtons.setPadding(
 new Insets(5, 5, 5, 5));
 paneForRadioButtons.setStyle(
 "-fx-border-color: green");
 rbRed = new RadioButton("Red");
 rbGreen = new RadioButton("Green");
 rbBlue = new RadioButton("Blue");
 colorToggleGroup = new ToggleGroup();
 rbRed.setToggleGroup(colorToggleGroup);
 rbGreen.setToggleGroup(colorToggleGroup);
 rbBlue.setToggleGroup(colorToggleGroup);
 rbRed.setOnAction(e -> {
 if(rbRed.isSelected())
 text.setFill(Color.RED);
 });
 rbGreen.setOnAction(e -> {
 if(rbGreen.isSelected())
 text.setFill(Color.GREEN);
 });
 rbBlue.setOnAction(e -> {
 if(rbBlue.isSelected())
 text.setFill(Color.BLUE);
 });
 paneForRadioButtons.getChildren().addAll(
 rbRed, rbGreen, rbBlue);}
private BorderPane getMainPane() {
 mainPane = new BorderPane();
 createTextPane();
 createPaneForButtons();
 createPaneForRadioButtons();
 createPaneForCheckBoxes();
}

```

```
 mainPane.setBottom(paneForButtons);
 mainPane.setCenter(paneForText);
 text.setFont(Font.font("Times New Roman",
 FontWeight.NORMAL, FontPosture.REGULAR, 20));
 mainPane.setLeft(paneForRadioButtons);
 mainPane.setRight(paneForCheckBoxes);
 return mainPane;
 }

 @Override
 public void start(Stage primaryStage) {
 Scene scene = new Scene(getMainPane(), 400, 200);
 primaryStage.setScene(scene);
 primaryStage.setResizable(false);
 primaryStage.show();
 primaryStage.setTitle("Option");
 }

 public static void main(String[] args) {
 launch(args);
 }
}
```

## Appendix: Classes and Methods

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b><u>java.nio.file.Paths</u></b></p> <ul style="list-style-type: none"> <li>• static Path get(String)</li> </ul> <p><b><u>java.nio.file.Path</u></b></p> <ul style="list-style-type: none"> <li>• Path getFileName()</li> <li>• File toFile()</li> </ul> <p><b><u>java.nio.channels.FileChannel</u></b></p> <ul style="list-style-type: none"> <li>• FileChannel open(Path)</li> <li>• int read(ByteBuffer)</li> <li>• int write(ByteBuffer)</li> <li>• void close()</li> </ul> | <p><b><u>java.nio.ByteBuffer</u></b></p> <ul style="list-style-type: none"> <li>• ByteBuffer allocate(int)</li> <li>• ByteBuffer wrap(byte[])</li> <li>• void flip()</li> <li>• void clear()</li> <li>• void rewind()</li> </ul> <p><b><u>java.lang.String</u></b></p> <ul style="list-style-type: none"> <li>• String[] split(String)</li> <li>• byte[] getBytes</li> <li>• boolean matches(String)</li> </ul> |
| <p><b><u>BorderPane</u></b><br/> BorderPane()<br/> void setBottom(Node)<br/> void setCenter(Node)<br/> void setLeft(Node)<br/> void setRight(Node)<br/> <b><u>EventHandler&lt;T extends Event&gt;</u></b><br/> void handle(T event)<br/> <b><u>Label</u></b><br/> Label(String)</p>                                                                                                                                                                                                 | <p><b><u>Font</u></b><br/> static font(String family, FontWeight, FontPosture, size)<br/> <b><u>Button</u></b><br/> Button(String)<br/> void<br/> setOnAction(EventHandler&lt;ActionEvent&gt;)<br/> <b><u>Scene</u></b><br/> Scene(Pane, double width, double height)</p>                                                                                                                                       |
| <p><b><u>Pane</u></b><br/> Pane()<br/> ObservableList&lt;Node&gt; getChildren()<br/> <b><u>VBox</u></b><br/> VBox(int spacing)<br/> ObservableList&lt;Node&gt; getChildren()</p>                                                                                                                                                                                                                                                                                                    | <p><b><u>HBox</u></b><br/> HBox(int spacing)<br/> ObservableList&lt;Node&gt; getChildren()<br/> <b><u>ObservableList&lt;Node&gt;</u></b><br/> void add(Node)<br/> void addAll(Node...)</p>                                                                                                                                                                                                                      |
| <p><b><u>Stage</u></b><br/> void setResizable(boolean)<br/> void setScene(Scene)<br/> void setTitle(String)<br/> void show()</p>                                                                                                                                                                                                                                                                                                                                                    | <p><b><u>CheckBox</u></b><br/> CheckBox(String)<br/> boolean isSelected()<br/> void<br/> setOnAction(EventHandler&lt;ActionEvent&gt;)</p>                                                                                                                                                                                                                                                                       |
| <p><b><u>Text</u></b><br/> Text(double x, double y, String text)<br/> void setFont(Font)<br/> void setFill(Paint)<br/> <b><u>ToggleGroup</u></b><br/> ToggleGroup()</p>                                                                                                                                                                                                                                                                                                             | <p><b><u>RadioButton</u></b><br/> RadioButton(String)<br/> void setToggleGroup(ToggleGroup)<br/> boolean isSelected()<br/> void<br/> setOnAction(EventHandler&lt;ActionEvent&gt;)</p>                                                                                                                                                                                                                           |