Faculty of Science

Course:	CSCI 2020U – Software Systems Development & Integration
Due date:	March 16th, 2021
Grade:	30%
Submission:	via Canvas, **pdf (Part I) and zip (Part II) files** – paste short questions' answers into this document and save it as PDF;

General Instructions

- This is an **individual** submission.
- You are **allowed to ask the instructor or the TAs** clarification questions via **private messages on MS Teams**. However, no answers that will directly solve your midterm will be provided.
- You should **not** post questions using **public channels**, neither in official nor unofficial course platforms.
- You must **not** share your questions nor your answers to this assignment.
- Any form of academic misconduct detected will be treated accordingly to the university's procedures.
- You must name your midterm **PDF** file "csci2020umidterm-lastname-firstname-studentID"
- Preferably, use a different text color for your textual answers.
- csci2020umidterm-Mohamed-Omar-100751749

Student Name: Omar Mohamed

Student ID: 100751749

Part I: Short Answer Questions (10 pts)

1. (1.5 pts) Given the directory and files below:
   - GitHub
     - MyProject 1
     - MyProject 2
       - HelloWorld.java
       - HelloWorld.class
       - GoodNight.csv
       - .gitignore

   a. Write the content of the .gitignore file for **MyProject 2** to ignore compiled files.

   *.class
   *.csv
   *.jar
   *.war
   *.ear
   *.nar
   *.zip
   *.rar

   b. Assume you opened a terminal in the GitHub folder, write the command(s) you would use to start **MyProject 2** as a new GitHub project, add all the current content, and commit changes:

   git init
   git add --all
   git commit -m “message to GitHub”

2. (1.5 pts) Write a build.gradle file’s content to run a Java program **CalculateTaxes.java**, which has an external dependency on ‘org.json’.

```
plugins {
   id 'java'
      id 'application'
}

repositories {
   mavenCentral()
}

dependencies {
   implementation 'org.json’
}

application {
   mainClassName = 'csci2020u.CalculateTaxes.java'
}
```

3. (2 pts) Imagine you are a senior software developer in a company and you are asked to write a brief introduction to the company's best coding practices to newly hired coop students. Your introduction should:

- Explain what are coding best practices.
- Provide 2-3 examples based on Java as the programming language.
- Explain why it is so important that everyone on-board learns and applies these practices.

Write your introduction below.

It's important that everyone here should learn and apply these practices during your work due to reducing errors caused by each other's work and allows for us to have an efficient work environment. The projects that you work on will be consist and can be maintained easily as well as allow for each of you as employee s to understand it easily and change it to the project's needs.

Some coding Practices that are best suited in a work environment would be the following

- Use Yoda Conditions which is just making the constants go first in an condition statement. The reason why utilizing Yoda conditions is a good practice is due to it preventing a mistake where it assigns the value by accident rather than executing what the programmer wanted. Which specifically is when the programmer does = rather than == in a conditional statement.
    - Not using Yoda conditions (bad practice)

      ```
      If (var=2){
              … //assigns var to 2, instead of executing the condition
      }
      ```

    - Using Yoda Conditions (good practice)

      ```
      If(2=var){
              Var1=0; //checks condition properly
      }
      ```

- Always use open and closed curly brackets no matter what. Essentially whenever you have a for/do/while loop or an if/else/try condition, you add a curly bracket after your condition and at the end of it .The reason why is that its best suited for referencing and organization of the code. If the case were to be you add something to your loop or if/else condition that results in an error, backtracking and resolving the error would be a lot more painful to track down.
    - Example of not using open/closed curly brackets (bad practice)

      ```
      If(a>10){
          System.println("A is bigger than 10");
          //Unknown when its over when there's more line of code underneath
      ```

    - Example of using open/closed curly brackets (good practice)

      ```
      If(a>10){
              System.println("A is bigger than 10");
      } //Easy way to notice that if statement is over
      ```

- Always Check for Null. Essentially what you do is make an if statement whenever you are checking a function or a variable.The reason behind this is under the case of your coworker sending you bad data to input to your code. By checking for Null you get ahead of the errors by knowing if a part of the data is unavailable or just not there, stopping any NullPointerException errors to happen
    - Example of not checking for null (bad practice)

      ```
      executeFunc();
      ```

- Example of checking for Null (good practice)

```
If(var1 !=null && var2 !=null){
        executeFunc(var1,var2);
}
```

4. (2 pts) Using your own words, contrast creating UI using FXML with programmatically method. Are there any conceptual advantages, or disadvantages between the two when it comes to software architecture?
FXML easily can integrate with the scene builder to create the applications you have programmed, along with there being a lot less code to type due to the components needed to create your scenes being more compacted together. FXML allows for better organization for those components as well. Programmatically JavaFX allows for an easy transition due to it being an extension of Java, allowing for people who know how to code in Java to be able to create scenes without learning a new programming language. However, programmatically the components needed are defined and then combined to create the application/scene which in this case can result in more issues adjusting code on both ends. Along with this programmatically front end and back end are split off from each other allowing for a more in-depth access to both of them. Overall FXML has features that the programmatically method(JavaFX in this case) utilizes to create the layout of the scene. Although of this, FXML still has limitations when it comes to more complex cases where using a programmatically method would be more easier.

5. (3 pts) Considering the hierarchy Ontario/Region/City is used to report on daily new covid-19 cases. Give an example of how that data would be represented in the different formats:
***Note**. You can create random data for your example data, it also only requires 3-5 records to display the structure, you do not need to come up with a large amount of data.*

a. CSV

```
Ontario, Region, City, dailyCases
Ontario, Niagara Region, St Catherines,50.0
Ontario, Durham Region, Ajax,15.0
Ontario, York Region, Markham,65.0
```

b. JSON

```
{"Ontario":{
" Niagara Region": {
        "Region" : "Niagara Region"
        "City": "St Catherines"
        "dailyCases": 50.0 },

"Durham Region": {
        "Region" : "Durham Region"
```

```
            “City”: ”Ajax”
            “dailyCases”: 15.0 },

      ” York Region”: {
            “Region” : “York Region”
            “City”: ”Markham”
            “dailyCases”: 65.0 },

          }
      }
```

c. XML

```
<Ontario>
      <region>
            <regionName> Niagara Region <regionName>
            <City>St Catherines <City>
            <dailyCases> 50.0 <dailyCases>
      <region>

      <region>
            <regionName> Durham Region <regionName>
            <City>Ajax <City>
            <dailyCases> 15.0 <dailyCases>
      <region>



      <region>
            <regionName> York Region <regionName>
            <City>Markham <City>
            <dailyCases> 65.0 <dailyCases>
      <region>
<Ontario>
```

Part II: Programming Question (20 pts)

For this part, it's recommended you use the **CSCI2020U-Midterm-Basecode.zip** given to you as base template for a JavaFX application. You can choose your environment of choice for JavaFX, however, the base code is an IntelliJ project which you can adapt to your setup.

You will implement the necessary code to fulfill the action of each of the buttons shown in Figure 1. You are given specific instructions for each of the buttons (Animation, 2D Graphics, About). However, the **general instructions** are

1. You may change the Scene, or the root object of the main scene (as you prefer) in order to change the application UI for each of the buttons accordingly.
2. You main create as many (if any) additional classes as needed.
3. You must keep coding best practices while you design your solution.
4. You may create the UI components programmatically, or via FXML/CSS according to your preference.
5. All 3 buttons different UI must contain a "Back to Main" link, which if clicked on, will allow the user to navigate back to the original mainScene as you see it on the screenshot in Figure 1.

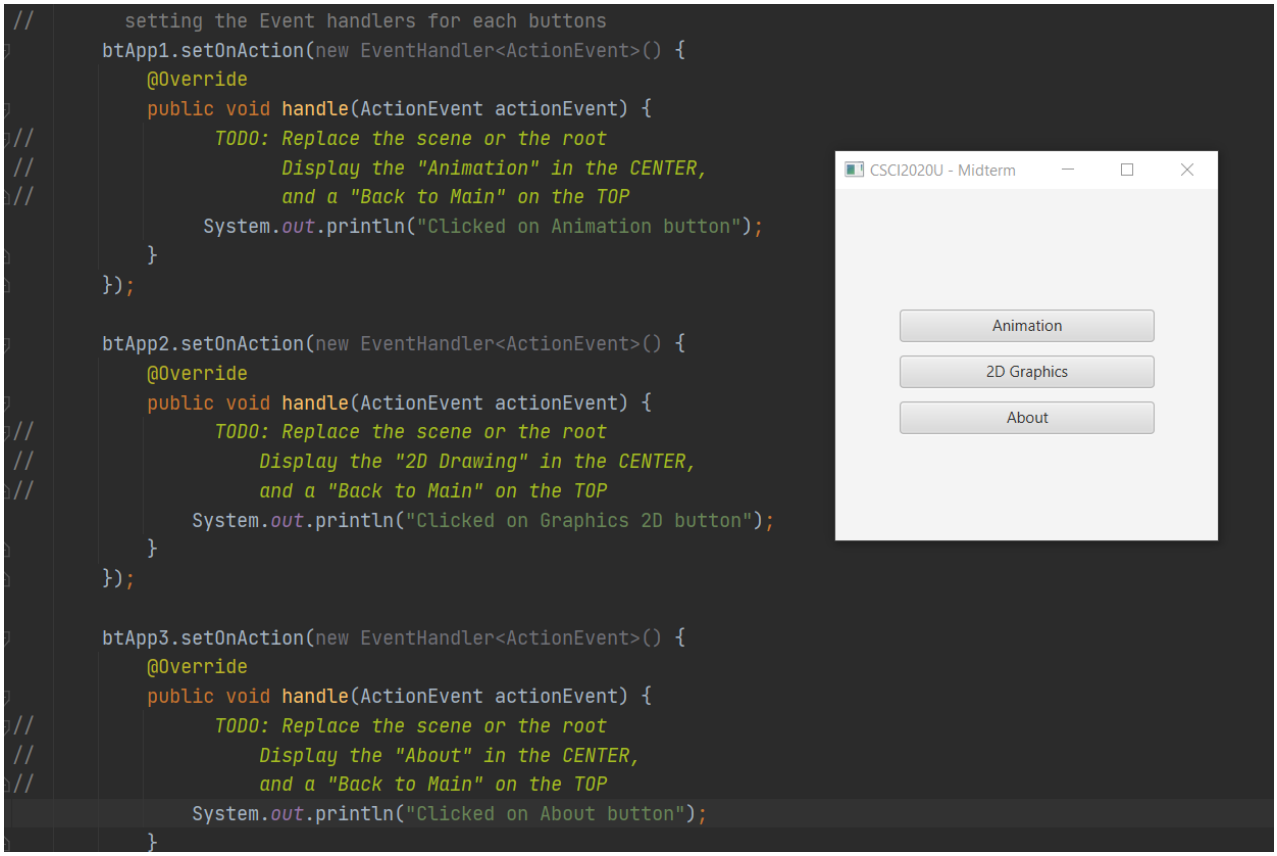

Figure 1. Event Handlers screenshot and the Main application UI.

The "Animation" Option

This UI will display an animation to the user using the ducks.png (Figure 2) sprite located in the "resources" folder. You **must not** alter the image, for example, image cropping.

- The sprite sheet has 4 types of ducks {wild, white, tan, grey} divided in columns {1-3, 4-6, 7-9, and 10-12} respectively.
- Each of the character actions are from left to right, and each row is a different view.

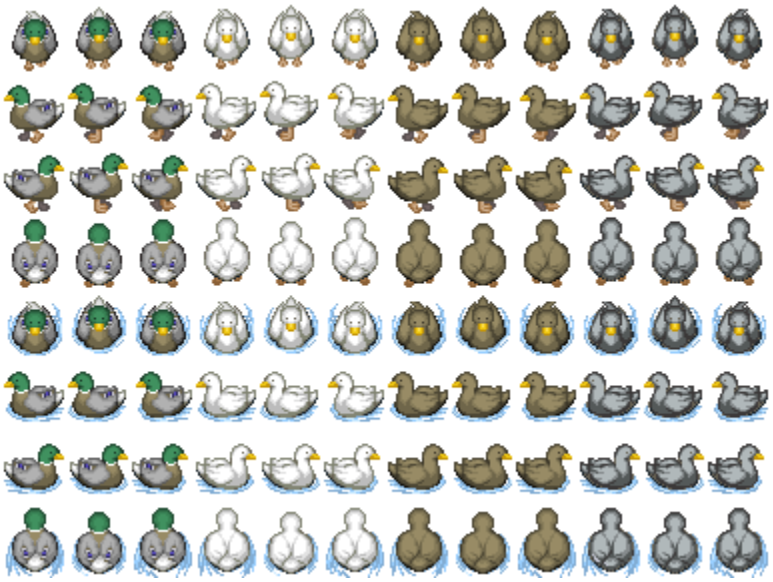

Figure 2. Source: http://www.rpgmakervxace.net/topic/2399-grannys-lists-animal-sprites/

Your task is to build an animation for **only 1 type of duck (see table below)**, where each row (view) will last 4 seconds and it will switch to the next view which will play for 4 seconds, and so on.
After all views were animated, you will loop back into the first view.

Last 2 Digits of your Student ID – OOOOOOOXX	Duck Type
00 – 25	Wild
25 – 50	White
50 – 75	Tan
75 – 100	Grey

The "2D Graphics" Option

Using 2D graphics, you will draw your name initials (letters) without using a **Text** or **SVGPath** objects, you must only use other basic shapes like lines, ellipses, rectangles, etc. You may also use properties like fill, stroke, etc.

Feel free to make it personal and express your personality with colors and typeface style. Besides the requirements of not using Text or SVGPath, the initials should be readable using the Roman alphabet as the baseline.

For reference add a label with your initials, so the grader can know which letters you are supposed to draw.

The "About" Option

This option will display information about you.
You may choose how to display the information (i.e. Text, Label, TextBox, Link).
The information displayed should be read from a XML file created by you in the "resources" folder following the template:\\

```
<info>
      <student id="">
            <name>  </name>
           <email>  <\email>
      </student>
      <software-description>
           Some description in text
      </software-description>
</info>
```

Part II Submission

You will submit 1 zip file with your project named "CSCI2020U-Midterm-LastName-FirstName-studentID.zip".

Submit your solution including

- the "resources" folder with your info.XML file
- all the .java files
- README.md file with instruction on "How to run" your program.

If you used IntelliJ you may save you project as .zip file (File/Export/Project as zip)