**ICS4U Final Project Assignment**



**Design and implement your own GUI Program**

**Introduction**

As a culminating task for the Grade 12 CS course in Java you will be building an application that showcases skills you have learned throughout the course. You will propose a project and then work, following a process/schedule, to accomplish the task. Project options can be anything from useful application software/tools to entertainment/games.

If you had to choose, what type of program would you make?

I want to give you an opportunity to experiment with the different types of GUI components that Java Swing provides. We’ve already seen JButtons, JTextFields, and JLabels; but those are relatively simple. Java Swing also has “Radio Buttons”, CheckBoxes, Sliders, Lists, Drop-downs, Progress Bars, etc. You can even experiment by having multiple Windows (pop-up Dialog boxes, etc.). For the final task of this course, I want you to design and create a GUI Program for the context of your choosing. I am not going to specify what your program must do (that’s up to you to choose). You can make a simple game like Snake / Minesweeper / Guessing game, you can make an application that performs simple calculations (like a financial calculator for mortgages), you can make anything you want (within reason).

Please don’t make your program too complicated. You will only have a few weeks to do this from scratch, and during the final few weeks of the semester our classes are 75 minutes.

This final task should be done in stages:

**1. Investigation:** You need to learn about the possible options for GUI components so that you know what you are capable of coding and what limitations you must work within. This link is a good starting point, but it gets very complicated as you go through each component: <https://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>

You can create a new NetBeans / Eclipse WindowBuilder project with an empty frame and use the drag and drop list of components to create sample components. Then you can run the program with no app logic to see what they look like and how the user interacts with them.

**2. Design:** You need to have a design in place before you can start to create components and code the app logic. I would like to see this design so that I can give you feedback on it in case something needs revision before you start your implementation.

Choose a program concept and create a “blueprint” of your GUI in the middle of a blank page. Around your blueprint include “blurbs” to say what type of component you are using for each part and what should happen when the user interacts with it. If needed, use arrows to connect the blurbs to the components on the blueprint. The only requirement I have is that your design must use at least 3 new types of components (in addition to labels, text fields, and buttons if you choose to use them).

Once you have a design drawn, take a picture of it and attach it to the Assignment in Google Classroom. Please make sure that all text is legible. Also, attach a Google Doc with a high-level description of your application (the smaller details should be on the blueprint). I will provide the feedback as soon as possible.

**3. Implementation:**

a. Create the GUI front-end according to your design. If you are using any images in your GUI, please use relative paths and not absolute paths (eg. no “C:\users\triantafillou\...” pathways should exist when using images). You can place any images you need inside of your Eclipse project folder and use relative paths to access them. This way when I download your program files (java files and images), I can run it without having to modify your code.

b. Code the application logic to give functionality to the components used. This part may require a lot of coding. I suggest you create a new class to be the “brain/memory” of your game/app and create an instance object of this class in your GUI class as a private attribute (global variable). Then you will be able to interact with it from anywhere inside of your GUI’s code.

**4. Testing:** make sure your program works according to your design and intended outcomes. Write a very short summary report (at most 2 pages) showing some screenshots of your program in use and listing possible test cases that you may have used to verify that your program works.

**5. Documentation:** make a “how-to” document that will explain to anyone using your program how to interact with it. This document should not be technical in nature and should be targeted to a general user.

Friday, January 7th:

• Design Blueprint document

The final product is due on Sunday, January 23rd.

• Implementation (java files + image files)

• Testing document

• “How-To” document

• Video Demonstration

Marking Scheme: 90/90

• 16/60 Application: G.U.I. “front end”

• 20/60 Thinking: App logic “back end” to make GUI functional.

• 8/60 Communication: Testing Document

• 8/60 Communication: “How-To” Document

• 8/60 Communication: Video Demonstration

Total 60/60 -> 30/30

Good luck and have fun!

Mr.T