ICS4U Python Unit 2 Review Project

For the review project in this course, you are to create a piece of useable software. You are to work individually on this project using any and all resources at your disposal, but the work must be your own. Any cases of plagiarism in any way will result in a mark of zero.

The following programming concepts must be used **correctly** and **appropriately** in your program:

- Variables & Constants
- Input & Output
- Selection (if & switch)
- Arrays
- Arithmetic Operators & Expressions
- Formatting of output (e.g. rounding, spacing)
- Repetition (counted & conditional loops)
- Functions and Modules

Try to make your program modular. Alternatively, you can also use Tkinter to implement

graphical solutions for your program.

Make sure to choose a program that meets your capabilities. Though the instructor will offer some help during the project development, it is expected that the student fully completes the coding themselves.

You should strive to use as many concepts as possible to show your understanding and ability to apply them in an application.

- Brainstorm program ideas (search online)
- When you have selected a final topic obtain approval from the teacher for that project.
- **Your project must be approved by the teacher**.

Step 1: Project Proposal & Planning (outline the program requirements/specifications)

Submit a **100 words page** written report with the following sections (sub-titles):

1. **Program Description (Problem Definition)**
   - detailed description of the application (program) that you plan to be create; include a list of variables.
2. **Programming Concepts**
   - outline the programming concepts you plan to use (e.g. if, switch, for, while, do-while, arrays, files, other topics you plan to research yourself, etc.) and how you plan/intend to use them.
   - describe/explain how you plan to use these concepts in your application.

Step 2: Coding and Documentation

- Begin coding (implementing your program) in Python.
- Include detailed internal documentation (comments) in your code.
- Each function should have comments describing the purpose, parameters, and any return values.
- Document the program all along the way (include comments) – NOT only at the end of coding.
- Ensure that all proper documentation and programming conventions are followed (e.g. spacing, indentation, naming of identifiers – variable and function names).

Step 3: Testing, Maintenance, User Guide

- Create a small **(100 words)** **User Manual or User Guide** for the project that explains how to use the application and its features. This will a Word file. docx and will be uploaded together with the project on the Hub.

Projects Ideas:

Games involving dice, cards, letters, and words provide good candidates. The following list provides some examples of games that are suitable for this project.

- Yahtzee
- Hangman / Wheel of Fortune
- Jeopardy
- Trivia/Quiz Game
- Pig Dice
- 7/11
- Mastermind
- BlackJack

Work at your level.

Some games or concepts will require additional research beyond what you have covered in the course. Some students may be interested in exploring graphics. Other games are well-suited to the use of two-dimensional arrays. It is important that you select a program/game that is within your capabilities.

Some additional candidate games include:

- Battleship (graphics)
- Tic-Tac-Toe
- Checkers / Chess
- Minesweeper
- Tower of Hanoi

Submit on the Hub:

1. **Project Proposal & Planning (100 words)**
2. **Coding and Documentation**
3. **User Guide (100 words)**

A **Marking Rubric** is provided.

Name: ICS4U Project Marking Rubric ____ / 100

THINKING – Program Analysis & Design

Level 0 (<50%)	Level 1 (50 – 59%)	Level 2 (60 – 69%)	Level 3 (70 – 79%)	Level 4 (80 – 100%)
Program design proposal not provided. No programming concepts outlined or explained. Implementation steps not followed.	Program design proposal is minimal. Few programming concepts outlined and explained. Implementation steps minimally followed. Little to no analysis & problem-solving skills shown.	Program design proposal has some missing parts/steps Some programming concepts outlined and explained. Some implementation steps were followed. Some analysis & problem- solving skills shown.	Program design proposal is clear and complete (includes all major features/ steps). Most programming concepts outlined and explained well. Most implementation steps were followed. Considerable analysis & problem-solving skills shown.	Program design proposal is clear, complete, concise, comprehensive, and modular. All programming concepts outlined and explained thoroughly. All implementation steps show excellent analysis & problem-solving skills.

_____ / 20

APPLICATION – Program Implementation & Functionality, Code Structure & Design

Level 0 (<50%)	Level 1 (50 – 59%)	Level 2 (60 – 69%)	Level 3 (70 – 79%)	Level 4 (80 – 100%)
The program is not provided.	Incomplete. Program is missing some major features and/or has significant errors. Demonstrates few required programming skills & concepts.	Somewhat complete. Program is missing some minor features and/or has some minor errors. Demonstrates some required programming concepts & skills.	Mostly complete. Program has some minor errors and/or omissions. Demonstrates most required programming concepts & skills.	Complete and passes all tests with no errors. Demonstrates exceptional understanding of all programming concepts & skills. Exceeds expectations.

_____ / 60

COMMUNICATION – Code Readability, Organization, Documentation & Maintenance

Level 0 (<50%)	Level 1 (50 – 59%)	Level 2 (60 – 69%)	Level 3 (70 – 79%)	Level 4 (80 – 100%)
No formatting or comments provided. Methods/variables not used or naming incorrect, not descriptive. Documentation is not provided.	Minimal formatting and comments provided. Method/variable naming incorrect, not descriptive. Documentation is minimal/incomplete. Many spelling and/or grammar errors interfere with understanding.	Some formatting and comments provided. Method/variable naming somewhat correct, descriptive. Documentation is missing some parts. Some spelling and/or grammar errors interfere with understanding.	Good formatting and comments provided. Method/variable naming mostly correct, descriptive. Documentation is mostly complete. Minor spelling and/or grammar errors do not interfere with understanding.	Excellent formatting and comments provided. Method/variable naming meaningful and descriptive. Documentation is exceptional and detailed. No spelling or grammar errors.

_____ / 20