Assignment 4

Due: After Unit 10
Total: 100 marks
Weight: 20% of your final grade

For this last assignment, you are required to complete **one** of the three projects listed below. Complete and submit it after you have finished working through Unit 10.

Your submission must include the following files:

- (**10 marks**) Your assignment report
- (**5 marks**) The program files you developed for the project you chose for this assignment, the Jupyter Notebook files you created and used during your study of Unit 10, and the Jupyter Notebook you created and used when working on this assignment

Format

Your assignment report should be in either (a) Markdown format (.md), written in VS Code, or (b) Word format. When you create a .md file in VS Code, VS Code recognizes it as a Markdown document and you can then use all the Markdown tags to format the content. For Assignment 4, name the report **assignment-report.md** or **assignment-report.docx**. Place it in a subfolder called **assignment4** within your main folder called **comp218**.

The assignment report must begin with a cover page stating the course number and name, the assignment number, your name and student ID, and your best estimate of your time spent on the assignment. The cover page will then be followed by your documentation for work on the project. Your documentation should include

1. your interpretation of the project;
2. an analysis of the tasks and requirements, techniques, and/or algorithms you used to solve the problems or accomplish the tasks;
3. an explanation of modules, classes, and functions you wrote and/or used in the system developed for the project; and
4. a brief user guide.

When you submit the assignment, all the files for the assignment, including the Jupyter Notebook files, must reside in the folder named **assignment4**. Compress the entire **assignment4** folder into a file named **assignment4.zip** and submit the zip file to the Assignment 4 drop box on the course home page.

Project 1: GUI-Based Course Management System (85 marks total)

Study relevant materials available in the course or on the internet and try out class definitions in Jupyter Notebook within VS Code when answering the following questions.

1. (**10 marks**) Define a student class modelling a student, including name, student ID, start date in the class, name of tutor assigned, and a list of assessment records. Each assessment record will be a tuple containing a number to identify the assessment, the weight of the assessment, and the mark, which should be zero for a newly added instance to a course.

2. (**10 marks**) Define a course class modelling a course, including the course number such as comp218, title, revision number, date of initial offering, and list of students, course professor and assessment schedule as a list of assessment items. Each assessment item is represented as a tuple of (ID, name, weight) in which the ID is the assessment item ID; name is the assessment name, such as Assignment 1; and final exam. Weight is the percentage of the assessment towards the final grade.

3. (**55 marks**) The GUI application should meet the following criteria:
   a. There should be a button for adding a new course to the system, which will open a form for input and save basic course info previously metioned, with the list of students empty. Note that when saving the basic course info, the system should be able to check whether the total weight of all assessment items make up 100%.
   b. When a new course is added to the system, a unique binary file will be created for permanent storage of the course data.
   c. At the start of the system, the system should automatically load all courses from their respective binary files to restore their internal object representation.
   d. There should be a button to get a list of courses being offered to students.
   e. The user should be able to select a course from the list.
   f. The user should be able to add students to the selected course.
   g. The user should be to see a list of students in the course.
   h. The user should be able to select a student from the list.
   i. The user should be able to record an assessment for the selected student.
   j. The system should automatically calculate display the final grade of the student for the course.
   k. The user should be able to see a list of assessments, including the calculated final grade for the selected student.
   l. There should be a button to shut down the system, but before shutting down the application, the system must save/pickle the data for each course back to its binary file.

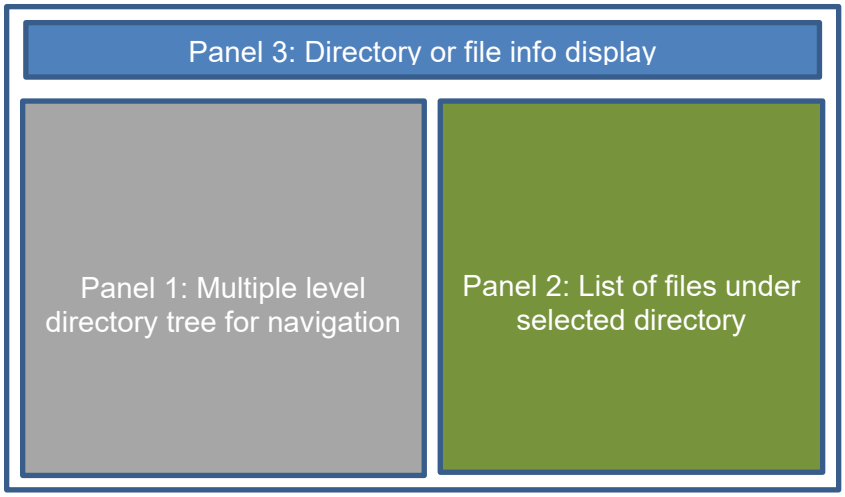
4. (**5 marks**) Your analysis and design of the system should be well documented in your assignment report.

5. (**5 marks**) Within each of your program files, there should be a docstring at the beginning stating the name and purpose of the file, as well as the ownership and revision history. One docstring is required for each class and function/method class. End-of-line comments are desired when deemed necessary.

Project 2: GUI-Based File Explorer (85 marks total)

Use the os and os.path module to develop a GUI-based file explorer similar to what you have used on your computers. The requirements are as follows:

1. (**15 marks**) The GUI must have three display panels, as illustrated below:



2. (**35 marks**) In Panel 1, a user can select any node in the directory tree by left-clicking the node with the mouse. If the node has subdirectories, the subdirectories at the next level will be displayed under the selected node at the next level of the tree structure, and a summary of the selected directory will be displayed in Panel 3. Otherwise, if the selected node doesn't have any subdirectory, the files under it will be displayed on Panel 2. The summary should include the full path to this directory, the number of immediate subdirectories, the number of files right under it, and the total storage space taken by all the files under it.

3. (**35 marks**) In Panel 2, when a user selects a file by left clicking the file name, the colour of the file name will change, and a summary of the file will be displayed in Panel 2. The summary should include the full path to the file, including the file name, the size of the file, and the date it was last modified.

Project 3: A GUI-Based Quiz System (85 marks total)

For this project, design and develop a quiz system with a graphic user interface (GUI). The requirements are detailed below:

1. (**10 marks**) Define a class named *Quiz_item*, which models multiple-choice questions and the correct answers. In additional to the constructor, it should have methods for changing the description of the question, individual choices, and the correct answer.

2. (**10 marks**) Define a class named *Quiz* that uses the Quiz_item class defined above to model a quiz that contains a number of quiz items that allow a user to create a quiz, add quiz items to the quiz, display the entire list of quiz items with correct answers, and execute a quiz on a user and calculate the score. At the end of the quiz, it should present to the user both the score and the correct answers

to incorrectly answered questions. The quiz items should be displayed to the user one by one during the quiz.

3. (**50 marks**) The GUI-based system should allow a user to do the following:
   a. Create a quiz
   b. Select a created quiz and add new quiz items
   c. Select a quiz and preview all the quiz items
   d. Select a quiz and execute the quiz by presenting the quiz items one by one
   e. Present to the user, at the end of the quiz, the score in percentage and the correct answers to the incorrectly answered question.

   The quiz items should be displayed to the user one by one during the quiz.

4. (**15 marks**) When executing a quiz, the system should have a timer so that the entire quiz is done within a given time. The timer must show the total number of quiz items in the quiz, the number of items remaining, and the amount of time allocated for the entire quiz as well as for the remaining quiz items.