

Computational Thinking and Problem Solving (COMP1002)

Mini-Project (20%)

(Due **at noon** on 2nd December 2022 (Friday))

In this project, form among yourself **a group of 3 - 4 members**. Enroll yourself to one of the groups in Blackboard. Nominate one group member as the group leader. Develop a simple Python-based system to solve a real-life problem. Choose ONLY ONE of the following topics:

Topic 1: Personal Expense Tracker

Having a good spending habit is essential to maintaining a healthy personal financial status. To facilitate the process, a personal expense tracker may help, which is an application that help users to keep track of their income and spendings, as well as allocating the money systemically for various purposes (e.g., saving, investment and charity). You are going to use Python to implement a Personal Expense Tracker application. Your program allows its user to input their income(s), spendings and other allocations. Here are the compulsory functions of the system:

- i. Input/store/update/retrieve the income(s), spendings and other allocations
- ii. Display the total income of last year
- iii. Display the top three spendings in the past 30 days

Topic 2: Computer Hardware Comparator

A PC consists of a number of hardware components, e.g., CPU, display card, RAM, hard drives and motherboard. Each of the components has various attributes (metrics) that determine its performance. For example, a CPU has its clock speed and size of caches, while a hard drive has its size and its read/write speed. You are going to create a Python program, to determine the overall performance, in terms of a numeric score, of a PC, with certain hardware configurations. You may define your own metrics to determine the score of a particular hardware component and sum them up to become the final score of the PC. Here are the compulsory functions of the system:

- i. Input/store/update/retrieve the name, description, performance attributes of a hardware component
- ii. Determine the numeric score of a particular hardware component based on your defined metrics
- iii. Customize a PC based on the available hardware components information stored in the system and provide the overall score of that PC

Topic 3: Restaurant Point-of-Sale (POS) System

A traditional restaurant is going to digitalize its service. You are asked to develop an information system for a restaurant to manage its menu and order. Here are the compulsory functions of the system:

- i. Input/store/update/retrieve the information on the menu configuration, meal sets, price, order quantities, date, and payments
- ii. Calculate the total price once an order is made
- iii. Review the sales information by day, week, month, year, or a certain period
- iv. Identify the best and least sales of meal sets

In your chosen topic, propose and implement at least ONE extra function. Justify your proposal in the report.

No database system is needed for this project. Use text file(s) to store your data persistently.

Your system is to be implemented as a console program. Note that it is NOT compulsory and NOT required that your program has a graphical user interface.

Deliverables

1. A report in **.pdf** format documenting the process of solving the problem.
 - a. Five pages maximum.
 - b. The report must contain the following information and sections. There is no need to include the code, because it is in the **.py** file(s).
 - i. Your group number, member names and student IDs
 - ii. The problem description
 - iii. Data abstraction (including a description of the key data types and representations, perhaps via examples)
 - iv. A Python implementation of the data types
 - v. A modular design of the program via the definition and use of a few key functions/procedures
 - vi. Any special observation or approach you would like to highlight
2. A well-documented Python program in a **.py** file(s).
 - a. You must use *docstring* to describe each function in the form of comments.
 - b. By using *appropriate* comments and variable names, your program must be easy to follow and understand.
 - c. Give proper *reference* to the source of the code that you adopt for your program.

Submission Instructions

Follow the steps below:

1. Create a folder and name it as **G<group no>**.
e.g., **G11**
2. Name the **.py** file as **G<group no>.py**,
e.g., **G11.py**
If there are more than one source files, the one with the *main()* function (i.e., the entry point of your program) should be named as required above.
3. Submit your report in a **.pdf** file. Name the single **.pdf** file as **G<group no>.pdf**,
e.g., **G11.pdf**
4. Put all the **.py** and **.pdf** files into the folder created in Step 1.
5. Compress the folder (**.zip**, **.7z**, or **.rar**).
6. Only the group leader submits the compressed file to Blackboard.

A maximum of **3 attempts** for submission are allowed. **Only the last attempt will be graded.** A late penalty of 5% per hour will be imposed.

Any wrong file naming and submission will be given ZERO mark. If your program cannot be run successfully (i.e., having any syntax error(s)), ZERO mark will be awarded for the program, regardless of how much you have coded.

All work must be done on your own. Plagiarism is serious offence. Any plagiarism cases (both copier and copier) will be given ZERO mark plus a deduction of the maximum mark of this assignment. Serious cases would be submitted to the Student Discipline Task Group (SDTG) of the department for further disciplinary actions.

Assessment Criteria

This project is assessed based on the following rubrics:

	Excellent	Good	Satisfactory	Weak	Fail
Report (40 marks)	Identify all requirements and data abstraction, and provide a comprehensive system design and sound justifications. (31-40 marks)	Identify most of the requirements and data abstraction, and provide good system design and justifications. (21-30 marks)	Identify some of the requirements and data abstraction, and provide system design and justifications. (11-20 marks).	Identify a few of the requirements and data abstraction, and provide system design and justifications. (1-10 marks)	No requirements, data abstraction, system design and justifications are provided. (0 mark)
System Implementation (50 marks)	Develop an application that satisfies all the system requirements and well documented. (41-50 marks)	Develop an application that satisfies most of the system requirements and properly documented. (31-40 marks)	Develop an application that satisfies some of the system requirements and fairly documented. (15-30 marks)	Develop an application that satisfies a few of the system requirements and documented. (1-15 marks)	Unable to develop an application that satisfies the system requirements. (0 mark)
System Usability (10 marks)	The system provides an excellent user experience. (9-10 marks)	The system provides a good user experience. (6-8 marks)	The system provides a fair user experience. (3-5 marks)	The system provides an unsatisfactory user experience. (1-2 marks)	The user interface provides no clue to the system functions. (0 mark)