Computational Thinking and Problem Solving (COMP1002) and Problem Solving Methodology in Information Technology (COMP1001)

Course Project
(Due at noon on 1st December 2021)

In this project, your group (3 - 4 members) is required to develop a simple Python-based system to solve a real-life problem. Choose ONLY ONE of the following topics:

<u>Topic 1: Password Management System</u>

Nowadays, we have accounts in numerous online applications including Email, social networking, banking, shopping, etc. Password is set to secure these accounts. A good practice is that no password should be reused in multiple accounts. However, most people are often unable to recall their passwords and so they may write them down on a piece of paper. You are going to use Python to implement a Password Management System. Your program allows its user to manipulate his/her password(s) stored in the local file system. Here are the compulsory functions of the system:

i. Password input/storage/update/retrieval
ii. Random Password Generation
iii. Password Strength Analysis (e.g., weak/medium/strong)

<u>Topic 2: Vaccination Tracking System</u>

To attend face-to-face classes in PolyU during the pandemic, students and staff members have to be vaccinated. You are going to use Python to implement a Vaccination Tracking System. Your program allows students and teaching staff to input vaccination information and they are stored in the local file system. There is a role called administrator, who is able to manipulate the vaccination information in the system. Here are the compulsory functions of the system:

i. Vaccination information input/storage/update/retrieval
ii. Administrator login
iii. Vaccination Information Analysis (e.g., number/percentage of (non-)vaccinated students/staff, names/department of (non-)vaccinated students/staff)

<u>Topic 3: Workshops Enrollment System</u>

COMP offers workshops regularly for the PolyU students. All students who feel interested enroll the workshops and will be notified whether the enrollment is successful. The enrollment allocation is usually on a first-come-first-served basis. You are required to use Python to implement a Workshops Enrollment System. Here are the compulsory functions of the system:

i. Workshop information input/storage/update/retrieval by administrator
ii. Workshop enrollment/cancellation by students
iii. System login by administrator and students

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 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.
        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 single `.py` file.
    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>_<your names>, e.g., **`G11_CHANTaiMan_WongTaiSin_LeeSimon`**
2. Submit the source file (**`.py`**). Name the **`.py`** files as **G**<group no>**`.py`**, e.g., **`G11.py`**
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.
5. Compress the folder (**`.zip`**, **`.7z`**, or **`.rar`**).
6. Submit 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 copiee) 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)