NGEE ANN POLYTECHNIC

Programming 1 (PRG1)

Year 1 (2023/24), Semester 1

SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Cyber Security & Forensics
Diploma in Data Science
Diploma in Immersive Media
Diploma in Information Technology
Common ICT Programme

ASSIGNMENT

Due on 13 August 2023 (Sunday), 2359 hours

Weightage:	30% of Module
Individual/Team/Both:	Individual
Format:	Programming and Presentation Basic Requirements (55%) Advanced Requirements (25%) Proper Documentation, Programming Style and Presentation (20%)

Penalty for late submission:

- 10% per day from the due date.
- NO submission shall be entertained after 7 calendar days of the due date.

There is a total of 8 pages (including this page) in this handout.

WARNING

If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action will also be taken.

Similar action will be taken for the student who allows other student(s) to copy his/her work.

1. OBJECTIVE

This assignment assesses the student's ability to apply relevant programming concepts to develop a simple application using Python programming language.

2. SCOPE

A simple application is to be developed for the user to process the HDB carpark availabilities.

You are assigned to develop a simple Python program to demonstrate the features provided by the system.

You are given THREE data files:

(a) carpark-information.csv

- The first row contains the column heading (Carpark Number, Carpark Type, Type of Parking System, Address).
- The rest of rows contain the actual data of the information of all the HDB carparks.

Figure 1 shows the partial data of the file open in Notepad.

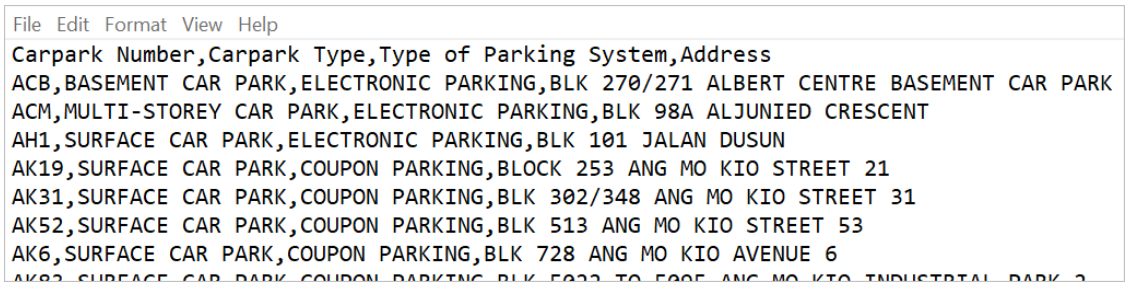

Figure 1 – partial content of data file "carpark-information.csv"

(b) carpark-availability-v1.csv and **carpark-availability-v2.csv**

- These files captured the lots available for all HDB carparks at two different period.
- The first row contains the timestamp of when the data was captured.
- The second row contains the column heading (Carpark Number, Total Lots, Lots Available).
- The rest of rows contain the actual data of the various car park availability.

Figure 2 shows the partial data of the file open in Notepad.

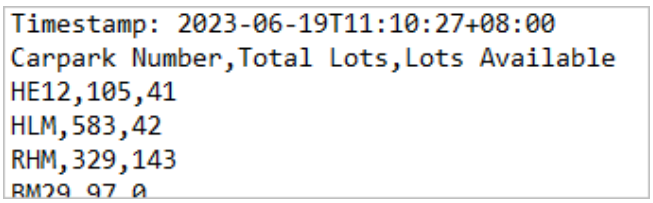

Figure 2 – partial content of data file "carpark-availability-v1.csv"

The assignment consists of "**Basic Requirements**" and "**Advanced Requirements**" as described in sections 3 and 4. You MUST complete the basic requirements BEFORE proceeding with the advanced requirements.

For this assignment, you are expected to:

- Understand the problem completely and plan your program layout before you start coding your program;
- Develop the solution for each task by using function;
- Functions develop should be as generic as possible - values used in functions should be passed in as the function parameters;
- Implement and test each feature as it is developed;
- Use Lists and Dictionaries;
- Use global variables sparingly;
- Do all the relevant data validations;
- Run your program with the three given data files.

3. BASIC REQUIREMENTS

The application should provide the following **basic** features:

- **Read data from 'carpark-information.csv' at the start of program execution**

  The program should read the data from the data file, 'carpark-information.csv' at the start of program execution and store the data in a list, each element in the list is a **dictionary** of the information of each carpark.

- **Display main menu** (and allow for repetition)

  When the program is executed, it should display the main menu as shown in Figure 3. When a user enters an option from 1 to 7, the program will process the option accordingly. After the option has been processed, the program will display the main menu again and the process is repeated until the user enters the option 0 to exit.

```
MENU
====
[1]  Display Total Number of Carparks in 'carpark-information.csv'
[2]  Display All Basement Carparks in 'carpark-information.csv'
[3]  Read Carpark Availability Data File
[4]  Print Total Number of Carparks in the File Read in [3]
[5]  Display Carparks Without Available Lots
[6]  Display Carparks With At Least x% Available Lots
[7]  Display Addresses of Carparks With At Least x% Available Lots
[0]  Exit
Enter your option:
```

Figure 3 - Main Menu

- **Display Total Number of Carparks in 'carpark-information.csv'**

  This feature allows the application to display the total number of carparks in the data file 'carpark-information.csv' read at the beginning of the program execution, as shown in Figure 4.

```
Option 1: Display Total Number of Carparks in 'carpark-information.csv'
Total Number of carparks in 'carpark-information.csv': 2199.
```

Figure 4 – Total Number of Carparks in 'carpark-information.csv'

- **Display All Basement Carparks in 'carpark-information.csv'**

  This feature allows the application to display all basement carparks in the data file, 'carpark-information.csv' as well as the total number, as shown in Figure 5.

```
Option 2: Display All Basement Carparks in 'carpark-information.csv'
Carpark No Carpark Type       Address
ACB        BASEMENT CAR PARK  BLK 270/271 ALBERT CENTRE BASEMENT CAR PARK
BBB        BASEMENT CAR PARK  BLK 231 BRAS BASAH BASEMENT CAR PARK
BM29       BASEMENT CAR PARK  BLK 163 BUKIT MERAH CENTRAL
BRB1       BASEMENT CAR PARK  BLK 665 BUFFALO ROAD BASEMENT CAR PARK
. . . < e t c > . .
Y62M       BASEMENT CAR PARK  BLOCK 342 YISHUN RING ROAD
Total number: 39
```

Figure 5 – All Basement Carparks in 'carpark-information.csv'

- **Read Carpark Availability Data File**

  This feature prompts the user for the name of the data file, reads the data and store the data in a list, each element in the list is a **dictionary** of the information of each carpark. The timestamp in the first line of the data file is displayed as shown in Figure 6 (value underlined is the user input).

```
Option 3: Read Carpark Availability Data File
Enter the file name: carpark-availability-v1.csv
Timestamp: 2023-06-19T11:10:27+08:00
```

Figure 6 – Read Carpark Availability Data File

- **Print Total Number of Carparks in the Carpark Availability Data File**

  This feature displays the total number of carparks in the carpark availability data file read in option 3 as shown in Figure 7:

```
Option 4: Print Total Number of Carparks in the File Read in [3]
Total Number of Carparks in the File: 1931
```

Figure 7 – Print Total Number of Carparks in the Carpark Availability Data File

Note: User can only choose this option after option 3 is done.

- **Display Carparks Without Available Lots**

  This feature displays the carpark number of all carparks without available lots (i.e. lots available = 0) as well as the total number of such carparks, as shown in Figure 8.

```
Option 5: Display Carparks without Available Lots
Carpark Number: BM29
Carpark Number: Q81
Carpark Number: C20
. . . < e t c > . . .
Carpark Number: B65L
Total number: 109
```

Figure 8 – Display Carparks Without Available Lots

Note: User can only choose this option after option 3 is done.

- **Display Carparks With At Least x% Available Lots**

  This feature prompts the user for an input percentage and displays all carparks that has at least that many percent of available lots together with the total number of such carparks and the percentage calculated, as shown in Figure 9 (value underlined is the user input).

```
Option 6: Display Carparks With At Least x% Available Lots
Enter the percentage required: 95
Carpark No  Total Lots  Lots Available  Percentage
SK48                50              50       100.0
TBM2               243             242        99.6
B90                 75              75       100.0
TJ37               384             384       100.0
H87L               500             483        96.6
. . . < e t c > . . .
JS3L              2010            1978        98.4
Total number: 41
```

Figure 9 – Display Carparks With At Least x% Available Lots

Note: User can only choose this option after option 3 is done.

- **Display Addresses of Carparks With At Least x% Available Lots**

  This feature prompts the user for an input percentage and displays the **addresses** of all carparks that has at least that many percent of available lots together with the total number of such carparks and the percentage calculated, as shown in Figure 10 (value underlined is the user input). Some carpark number may not be available in the carpark-information.csv, in this case you are not required to display the address.

```
Option 7: Display Addresses of Carparks With At Least x% Available Lots
Enter the percentage required: 95
Carpark No  Total Lots  Lots Available  Percentage  Address
SK48                50              50       100.0  BLK 321 ANCHORVALE DRIVE
TBM2               243             242        99.6  BLK 73A TELOK BLANGAH STREET 32
B90                 75              75       100.0  BLK 35A BEDOK SOUTH AVENUE 2
TJ37               384             384       100.0  BLK 337 TAH CHING ROAD
. . . < e t c > . . .
JS3L              2010            1978        98.4  BLK 624A JURONG WEST STREET 61
Total number: 41
```

Figure 10 – Display Addresses of Carparks With At Least x% Available Lots

Note: User can only choose this option after option 3 is done.

- **Program validation**

  Add appropriate validation for the Basic requirements of the program.

- **Program documentation**

  The program should have sufficient comments, which includes your name, class, date as well as the description for each function.

4. ADVANCED REQUIREMENTS

The application should provide the following **advanced** features.

- **Display All Carparks at Given Location**

  This feature allows the user to enter a location, searches for all carparks that are at this location and displays the carpark number, total lots, lots available, percentage of lots available (in 1 decimal place) and the address. Output should be displayed neatly in a tabular form. It also displays the total number of carparks found or a message indicating no carpark found if the location is not found in the file.

- **Display Carpark with the Most Parking Lots**

  This feature displays the carpark that has the most parking lots. To make the output complete, all information of the carpark should be displayed.

- **Create an Output File with Carpark Availability with Addresses and Sort by Lots Available**

  The original data file for carpark availability does not contain the address of carparks. This feature creates another file, **carpark-availability-with-addresses.csv**. The file should contain same data as the carpark-availability file that you have read in option 3 but with the addresses of the carpark added as one additional column. Data in the file are to be sorted according to the lots available in ascending order. Display on the output screen a message indicating the number of lines written into file as well as the filename.

- **Additional features – up to 10 BONUS marks**

  You may gain up to 10 bonus marks if you implement additional features to improve the application. Depending on the complexity of the feature, you may be awarded different marks for the feature. The following are some suggestions. Feel free to devise your own additional features but please confirm with your tutor before-hand.

  - Use real-time carpark available data from data.gov.sg
  - Change the user-interface to Graphical User Interface

  Do ensure that you have all the basic and advanced requirement working before you work on your additional feature. Programs with and without additional features are to be saved in different files.

Note:

- ***You are NOT to do the program in Object-Oriented approach.***
- ***You are NOT allowed to use any external libraries for the basic requirements.***
- ***You should implement the advanced requirements only AFTER all the basic requirements have been fully implemented (and fully working).***
- ***You should implement the additional features only AFTER all the 3 advanced requirements have been fully implemented (and fully working). Save the program containing the additional features in a different file name and submit both programs.***
- ***You should think carefully what input is required for each option if there is any.***
- ***You should design your own output for the advanced requirements.***
- ***You should do all the possible data validation in order to score.***

- ***You are required to present your solution to your tutor. Your tutor may ask you questions to verify and assess your understanding of your work. Your tutor may ask you to make some changes to your program to handle another similar feature.***
- ***NO MARKS will be awarded for the advanced requirements if all the basic requirements have NOT been fully implemented (and fully working) or you are not able to show your understanding of the program during the presentation.***

5. DELIVERABLES

- Name the file `"S10009999_Assignment.py"` where `"S10009999"` is your student ID.
- If you have done any Additional feature(s), name the file(s) `"S10009999_Assignment_Extra1.py"`, `"S10009999_Assignment_Extra2.py"` … and state the additional feature clearly in the comment.
- Submit your program into **POLITEMall > Assignment > PRG1 Assignment Submission** by **13 August 2023, 2359 hours.**
- Present your application to your tutor during your PRG1 lessons the week after submission deadline (i.e. 14 to 18 August 2023).

6. ASSESSMENT

This assignment constitutes 30% of this module.

Performance Criteria for grading the assignment is as described below. Marks awarded will be based on **program code** as well as student's degree of understanding of work done as assessed during the **presentation**.

A Grade

- Program implements the Basic Requirements with all input validation successfully
- Program implements all 3 Advanced Requirements successfully
- Program demonstrates good design with the correct use of functions
- Program completes with good documentation
- Program has been tested adequately
- Excellent demonstration of program and showing excellent understanding of work done during the presentation

B Grade

- Program implements the Basic Requirements with input validation successfully
- Program implements some Advanced Requirements with some success
- Program demonstrates good design with the correct use of functions
- Program completes with documentation
- Program has been tested adequately
- Good demonstration of program and showing some understanding of work done during presentation

C Grade

- Program implements the Basic Requirements with some input validation
- Program implements one Advanced Requirements
- Program demonstrates good design with the use of functions
- Program completes with some documentation
- Program has been tested adequately
- Some demonstration of program and showing some understanding of work done during presentation

D Grade

- Program implements the *Basic Requirements* successfully
- Program completes with some documentation
- Program has been tested adequately
- Able to answer some question during presentation