

CSC1024 Programming Principles

Development of A Personal Book Management System using Python

1.0 Assignment Description

a) Assignment Scenario

As a book enthusiast, you have amassed a huge number of physical books in your personal library over the years that span across various literary genres. With your newfound knowledge of Python programming, you would like to use it to keep track of all the books in your collection. Hence, the main aim of this programming project is to develop a personal book management system using the Python 3 programming language.

2.0 Programming Requirements

a) Initial Data Preparation

The book management system will require you to prepare a text file containing all the initial data.

- **books_StudentID.txt** - Contains initial data of **at least 20 initial book information**. The information that should be tracked within the book management system is the *ISBN, author, title, publisher, genre, year_published, date_purchased,* and *status*. The information stored within the text file acts like a database containing all the book information.

Example of the text file:

- **books_StudentID.txt**

```
9781785785719,Yasha Levine,Surveillance Valley,Icon Books, Politics,2019,02-01-2020,read
9780241976630,John Maeda,How to Speak Machine,Portfolio UK,Artificial Intelligence,2019,02-01-2020,to-read
9781119055808,Andre De Vries,R For Dummies,John Wiley,Programming,2015,12-08-2016,read
9780062018205,Dan Ariely,Predictably Irrational,Harper Collins,Psychology,2010,03-06-2019,read
9780008327613,John Walsh,The Globalist,Harper Collins, Biography & Autobiography,2019, 11-05-2021,to-read
9780525538349,John Doerr,Measure What Matters,Penguin LCC,Management,2018, 21-09-2022,to-read
9780807092156,Viktor Frankl,Man's Search for Meaning,Random House,Philosophy,2019, 30-08-2021,to-read
```

b) Program Functionality

The book management system developed should provide the user with a menu of action items that could be executed repeatedly until the user explicitly chooses to 'Exit' the program. The options that should be provided to the user in the form of a main menu are as follows:

(a) **Add Book Record(s)**

- Add book information(s) into the system.
- Within this submenu, the user must be able to add information of single/multiple books.

(b) **Delete Book Record(s)**

- Delete book information(s) in the system.
- Within this submenu, the user must be able to delete information of single/multiple books.

(c) **Update/Edit Book Record(s)**

- Prompts the user for the *ISBN* or *author and title*.
- If there are matching records found, prompt the user to enter the new information for the book.
- Update the book record(s) according to the new information provided.
- Within this submenu, the user must be able to update information of single/multiple books.

(d) **Display**

- Display all the books that are currently in the system.
- The display should contain appropriate headings for each column.
- Data from **books_StudentID.txt** must be *read and stored in the program when the program first runs* so that all the book information previously stored in the text file will be displayed if it is the first option selected when starting the program.

(e) **Search for Book(s)**

- Prompts for the *ISBN, author, and title*.
- Display the information of a particular book using the information provided.

(f) **Exit**

- Stop the execution of the program.
- The program will write the updated data into **books_StudentID.txt**.

c) Programming Techniques, Logic, and Control Flow

In this programming project, you **must apply and demonstrate** efficient mastery of the following programming fundamentals in Python:

- Input and display data.
- Lists and file processing.
- Searching from a list.
- If control statement, and relational and logical operators.
- Loops.
- User-defined function.

Only the following modules are allowed in the development of the project (should you choose to use them):

- **random module (import random)** is allowed for generating random values.
- **datetime module** to work with date in Python.
- **os module** to clear the screen for better presentation.

3.0 Assessment Criteria

This programming project has a total of **100 Marks**. A brief overview of the assessment criteria and their allocated marks are as follows. [Kindly refer to Appendix A for the detailed marking rubric]:

Code Presentation [Total: 8 Marks]

- **Code Comment.** [Allocated Marks: 3 Marks]

This criterion assesses the details and informativeness of commenting on the computer source code with appropriate comments.

- **Identifier Names.** [Allocated Marks: 3 Marks]

This criterion assesses the appropriateness of the identifier names used in the program.

- **Code Readability.** [Allocated Marks: 2 Marks]

This criterion assesses the coding style adopted when writing the code.

Overall Program Functionality and Completion [Total: 12 Marks]

- This criterion assesses the book management system's operational and functional completeness as described in Sections 1.0 and 2.0 (a) and (b).
- The breakdown of the total allocated 12 marks is as follows:

Program Functionality	Allocated Marks
Add Book Record(s)	2
Delete Book Record (s)	2
Update/Edit Book Record (s)	2
Display	2
Search Book Record(s)	2
Exit	2

Program Technical Considerations [Total: 50 Marks]

- **User Interface/User Interaction.** [Allocated Marks: 10 Marks]

This criterion assesses the user interface design of the book management system's user-friendliness and ease of interaction.

- **Error Handling and Data Validation.** [Allocated Marks: 10 Marks]

This criterion assesses the error handling capability and data validation functionality of the book management system. The error handling capability and data validation functionality must be tested and clearly documented in the report.

- **Programming Techniques, Logic, and Control Flow. [Allocated Marks: 30 Marks]**

This criterion assesses the computational implementation and application of the programming techniques and fundamentals, as described in Section 2.0 (c). These programming techniques must be clearly explained in the report. The problem-solving techniques, clarity and preciseness of the explanation provided are the primary measurements for this assessment.

The breakdown of the total allocated 30 marks is as follows:

Programming Fundamental	Allocated Marks
Input and display data	5
Lists and file processing	5
Searching from a list	5
If statement, and relational and logical operators	5
Loops	5
User-defined function	5

Project Report [Total: 30 Marks]

- This criterion assesses the contents and quality of the report.
- Kindly utilize Appendix B provided as the Cover Page of your report.
- The breakdown of the total allocated 30 marks is as follows:

Report Component	Allocated Marks
Flowchart	10
Discussion of the programming techniques used	4
Sample test cases and outputs	4
Sample text file	2
Presentation. This is to assess each group member's contribution and understanding of the project (Each member must contribute to the programming task. Failure to contribute in terms of programming for the project will result in a substantial deduction of your overall marks for the previous components covered)	10

4.0 Submission Details

General Information

- This programming project has a total of **100 marks**.
- This programming project is a **final examination**.
- This programming project is a **group** project with each group consisting of 4 - 5 members.

Important Dates

- Distribution Date: **8th November 2023 (Wednesday)**.
- Submission Date: **11th December 2023 (Monday), before 6.00 PM**.
- **Late submission is not permitted.**

Appeal for late submission must abide by a valid reason, as described in the Sunway University Academic Regulations.

Submission Procedures

You are required to prepare the following materials for submission:

- A copy of the program source code with appropriate comments written to describe the programming techniques, program flow and logic, and user-defined functions in the restaurant management system.

Name the program source file as: **python_StudentID.py**

- A copy of the text file in .txt format that is used to store the relevant information.

Name the text file as: **books_StudentID.txt**

- A copy of the project report in PDF format. Name the report file as:
report_StudentID.pdf

The StudentID should be replaced with the Student ID of the group leader.

Zip all the files **python_StudentID.py**, **books_StudentID.txt**, and **report_StudentID.pdf** into a Zip file. Name the Zip file **GroupID.zip**.

The **group leader** is required to upload the **GroupID.zip** zip file to eLearn before the submission due date.

• THE END • HAPPY CODING •