

BSc (Hons) Computer Science
BSc (Hons) Cyber Security
Module Code: QH0305 Module Title: Problem Solving
Assessment Sheet 4

Instructions:

This is one of the eight assessment tasks which will contribute to the overall mark. You will need to complete the tasks as outlined below and then document them in a Word document file. As a minimum, you should provide screenshots of the following:

- Your code
- The output that your code generates

In instances where your code could generate different outputs depending on what values are given, you should provide multiple screenshots of the console screen, showing the different outputs in order to demonstrate that the code works correctly.

You must attempt all tasks on this sheet to achieve a higher mark. For example, if you want to gain marks between 70 - 100, you must complete all other tasks first and add them to your portfolio with screenshots.

A zip folder with all tasks must be attached inside of the portfolio (MS Word file).

Task: Library Management System

This assessment focuses on creating a **basic menu-driven Library Management System** for managing books which include Adding New Book, View Books and Search Books.

To achieve Marks (Between 40-49)

Requirements:

1. Implement a single menu option "**a. Add a new book**".
2. Prompt the user to input:
 - Book Title.
 - Author Name.
 - Book Genre.
3. Display the details of the book after it is added.
4. Use **only one book** in the system (no arrays or multiple books yet).
5. Provide screenshots of:
 - The code.
 - Outputs for different inputs.

Example Interaction:

Enter the book's title: The Alchemist

Enter the author's name: Paulo Coelho

Enter the book's genre: Fiction

Book added successfully!

Book Details:

Title: The Alchemist

Author: Paulo Coelho

Genre: Fiction

To achieve Marks (Between 50-59)

Requirements:

1. Complete all previous steps.
2. Add functionality to store **multiple books** (e.g., up to 5 books):
 - Use arrays to store book details (bookTitles, bookAuthors, bookGenres).
 - Allow users to add multiple books until they reach the limit.
3. Add a menu option to **view all books**:
 - **b. View all books.**
4. Keep the program running in a loop until the user chooses to quit.
5. Display all books in a list format when the user selects "View all books."

Example Interaction:

Menu:

- a. Add a new book
- b. View all books
- y. Exit

Choose an option: a

Enter the book's title: The Alchemist

Enter the author's name: Paulo Coelho

Enter the book's genre: Fiction

Book added successfully!

Menu:

- a. Add a new book
- b. View all books
- y. Exit

Choose an option: a

Enter the book's title: Harry Potter

Enter the author's name: J.K. Rowling

Enter the book's genre: Fantasy

Book added successfully!

Menu:

- a. Add a new book
- b. View all books
- y. Exit

Choose an option: b

Books in the Library:

1. Title: The Alchemist, Author: Paulo Coelho, Genre: Fiction
2. Title: Harry Potter, Author: J.K. Rowling, Genre: Fantasy

Menu:

- a. Add a new book
 - b. View all books
 - y. Exit
-

To achieve Marks (Between 60-69)

Requirements:

1. Complete all previous steps.
2. Add functionality to **search for a book by title**:
 - **c. Search for a book:**
 - Prompt the user to input a book title.
 - Check if the book exists in the library.
 - If found, display the book's details.
 - If not found, display: "Book not found."
3. Add **basic error handling**:
 - If the user enters an invalid menu option, display: "Invalid option. Please try again."
4. Make the menu loop **until the user selects "y" to exit**.

Example Interaction:

Menu:

- a. Add a new book
- b. View all books
- c. Search for a book
- y. Exit

Choose an option: c

Enter the book title to search: Harry Potter

Book found!

Title: Harry Potter

Author: J.K. Rowling

Genre: Fantasy

Menu:

- a. Add a new book
- b. View all books
- c. Search for a book
- y. Exit

Choose an option: z

Invalid option. Please try again.

Menu:

- a. Add a new book
- b. View all books
- c. Search for a book
- y. Exit

To achieve Marks (Between 70-100)

Requirements:

Complete all previous steps, then:

- All tasks must be accompanied by written descriptions or annotations. These must demonstrate satisfactory understanding of what the code does.
- To achieve higher grades, there is always a requirement of some independent research. Implement pointers in the program and explain their suitability. This should include how pointers improve the program's efficiency and flexibility in handling dynamic data structures.

Assignment Preparation Guidelines

- All components of the assignment report must be Word-processed (**handwritten text or hand drawn diagrams are not acceptable**), font size must be within the range of 11 point to 14 point including the headings, body text and any texts within diagrams.
- Standard and commonly used fonts such as Times New Roman, Arial or Calibri should be used.
- All figures, graphs and tables must be numbered and labelled with short explanations.
- Material from external sources must be properly acknowledged and cited within the text using the Harvard referencing system.
- All components of the assignment (text, diagrams, code etc.) must be submitted in one Word file.
- The report should be logically structured, the core of the report may start by defining the problem / requirements, followed by the proposed solution including a detailed discussion, analysis and evaluation, leading to the implementation and testing stage, finally a conclusion and a personal reflection on learning.
- Screenshots without description / discussion are not suitable as they do not express your understanding or support your work adequately.

Submission instructions

- This is a portfolio assignment with eight tasks in total. Each task will be completed and saved in the portfolio. Once the portfolio is completed, it should be submitted on Turnitin. The submission link to Turnitin can be found under the “Assessment Tab” in your module section in the SOL VLE.
- Please note file size limitation might apply. Your report must be under 250MB.
- The source code for each task should be **zipped** and **attached** to your Word document report submission in the appendix.
- The Assignment Brief can be found under the “Assessment Tab” in your module section in the SOL VLE.
- **Refer to the Assignment Brief** to find the links to Late Submissions, Extenuating Circumstances, Academic Misconduct, Ethics Policy, Grade marking and Guidance for online submission through Solent Online Learning (SOL).