UNIVERSITI TUNKU ABDUL RAHMAN

FHCT1022 PROGRAMMING CONCEPTS

FOUNDATION IN SCIENCE (Session 202205)

GROUP ASSIGNMENT GUIDELINES

ASSESSMENT: The assignment constitutes **30%** of the final grade

ASSIGNMENT THEME:
"Library System"

This assignment will assess the following Course Learning Outcomes (CLO) and Domain:

CLO	CLO Description	Domain
CLO 2	Develop computer programs by using various structures with relevant algorithms and functions.	Cognitive

1.0 Objectives

This assignment is designed to enable you to put into practice the knowledge and concepts you have acquired over the trimester on problem solving and Python Program development to solve a computing problem effectively.

A problem will be provided to you where you will need to employ the program development processes (Six (6) Steps of Program Development) to assist you to build a viable system. These six (6) steps include *defining the problem*, *developing an algorithm*, *testing the algorithm*, *coding the algorithm*, *running the program*, and *documenting & maintaining the program*. [For more details, refer to Lecture Topic 1.]

You are expected to fully cooperate with your team members to take a detailed study of the problem (domain knowledge), decompose a complex problem into smaller, manageable parts so that the entire team can work hand in hand to build efficient and effective modules and functions that can be integrated easily to form an integral solution.

You are expected to inculcate the needed level of problem-solving, coding skills as well as to acquire all related soft skills needed in solving a computing problem in a team at the end of this assignment.

2.0 Assignment Rules

1. **Submission Date**: **Monday, 5 September 2022 before 12.00pm**

2. **Group size: 4 - 6** persons in a group.

3. **Submission: a zipped folder that is submitted to Google Form link (will be posted in Week 11)**. It must consist of:
   - ONE (1) report (.docx), and
   - ONE (1) or more source code (.py)

4. **Penalty:** No marks (0%) will be awarded to everyone in the group if **plagiarism** is found, **and** 10% of the entire assignment marks will be deducted each day for **late submission**.

5. **Important Notes to Every Team Member**:
   - Each of you **MUST** contribute to the completion of the assignment.
   - You are **responsible** for forming your own assignment group and choosing your own group leader.
   - You can create a group on MS Teams to discuss or do your assignment. Be **responsive** to other group members.
   - Exchange contact details (emails and phone numbers) and stay contactable.
   - Your lecturer has the right to deduct your assignment marks if you are found to be not playing an active role in completing the assignment.

3.0 Requirements / Output

1.0 **Requirements for report**:

- You are expected to prepare a **NEAT** and **LEGIBLE** report.
- You do not have to think of what you need to include in your report because you will be able to download the report template from WBLE.
- You do not have to do extra. Just follow what is required in the report template.

2.0 **Requirements for algorithm**:

- You can use any flowchart-drawing applications to draw your flowcharts.
- You are advised to construct well-documented algorithm(s) (flowcharts) showing all the detailed steps, in a correct manner, on how the entire system works.

3.0 **Requirements for Python Program:**

- You **may** opt to enhance the system to include special features and functions that can simulate actual operations of such system.
- Your program must be error-free and can produce the right output.
- Your program must be well-thought-out, with the use of the right combination of control structures and subprograms (self-defined functions).
- Your program must have well-presented screen interfaces with the ease to navigate from one screen to another.
- Your program must include data validation, along with, meaningful error messages to alert the user when he or she keys in invalid data.
- Your program code must be tidy and readable.
- You are advised to adhere to a good coding standard and style.

4.0 Problem / Question

You are required to develop a simple library system with the following functions:

- S**tore** a list of books or other reading materials in the library's repository/catalogue (book title, author, subject, etc.).
- **Search / Find** a book or other reading materials, and view the details as well as the availability of such material for loan.
- **Check out / Borrow** an item (maximum 5 items / books).
- **Check in /Return** an item.

The system should be able to simulate the above-mentioned functions well with the flexibility to repeatedly search, check in and check out reading materials until the user chooses to exit the system.

You are also reminded to conduct your own research to gain additional knowledge on how to enhance the system, build a system that is not only workable, but contain features and functions that closely resembles how real-life systems were to function. The following are some added features you may look into but not mandatory:

- Security access system for users who will need to authenticate to login.
- Account and membership maintenance (add, edit, delete records). Data may include membership type, annual fee and one-time deposit, entitlement (number of books, number of weeks), penalty, renewal, membership status (active, barred, deactivated), etc.
- Repository / Catalogue of different types and maintenance of accession number, barcode number, catalogue type (a to f above, etc.), title, author, internal reference number (e.g. QA 76.73.P98.S84 2014), ISBN number) of the following reading materials:
  a. Books
  b. Databases
  c. E-journals
  d. E-books
  e. E-newspapers
  f. Institutional Repository
- Penalty and Payment: Penalty log (fine for late return), payment date, time, amount paid.
- Search / Find functionality using subset of a string / substring of just part of the reference number or title or author or from any part of the reference number, title or author.
- Using text (.txt) files to store data (log, catalogue, payment transaction).

5.0 Grading (100%)

1. **Algorithm (Flowcharts) (20%)**
   Flowcharts show each step of the solutions in a detailed, clear, and precise manner. Flowcharts are drawn using the correct symbols, line arrows and labels.

2. **Program Specifications / Correctness (30%)**
   All parts of the program work correctly without errors. All basic specifications are met. Your program must produce all expected output accurately. Your program is written according to the steps depicted / represented in the flowcharts.

3. **Program Control and Subprogram Structures (Code Efficiency) (10%)**
   The order in which program instructions are performed should be carefully controlled, and programming problems are reduced to combinations of controlled sequences, selections, and repetitions (loop) in the most efficient manner. You **MUST NOT** to use a recursive function call in your program.

   Programs are organised into subprograms / self-defined functions and are self-contained (independent of other parts of the program) with well-defined data (locally declared variables) and operations.

4. **Interface Design and Flexibility in Navigating between & within Functions and Error Rectification (10%)**
   Clear instructions are provided on the screen for users to navigate through the functions / subprograms / menus / submenus of the system efficiently. Users are also guided with the understanding of what data and data structures to be entered to allow the system to process it into the expected output.

5. **Validation and Error Messages (10%)**
   Proper validation and control are placed to check for correct data type, data range, data length / size, and completeness depending on the use of data structures in the system. Error messages on what goes wrong must be clearly displayed to alert users. Users must be directed clearly on how to go about rectifying the problem.

6. **Documentation and Coding Style (Code Appearance) (10%)**
   Your Python Program is documented in a neat and tidy fashion. Proper header and meaningful comments (internal documentation) are provided accordingly in the program. Line spacing and spaces used in the statements are standardised to allow readability.

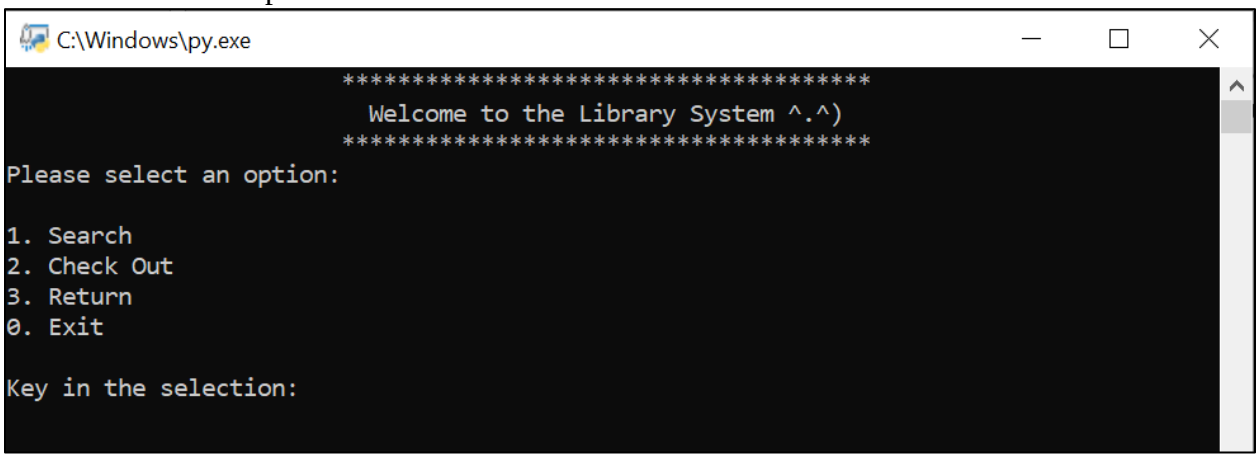
7. **Enhancement of System (10%)**
   Your program is enhanced to cater for features and functions that resemble those of a live (actual working) environment. It is also structured in such a way that it is configurable or adaptable to changes in requirements (minimum changes are required to be made to the program and structure when there is a requirement change) – having little "hardcoding" of any sort.

6.0 Screenshots

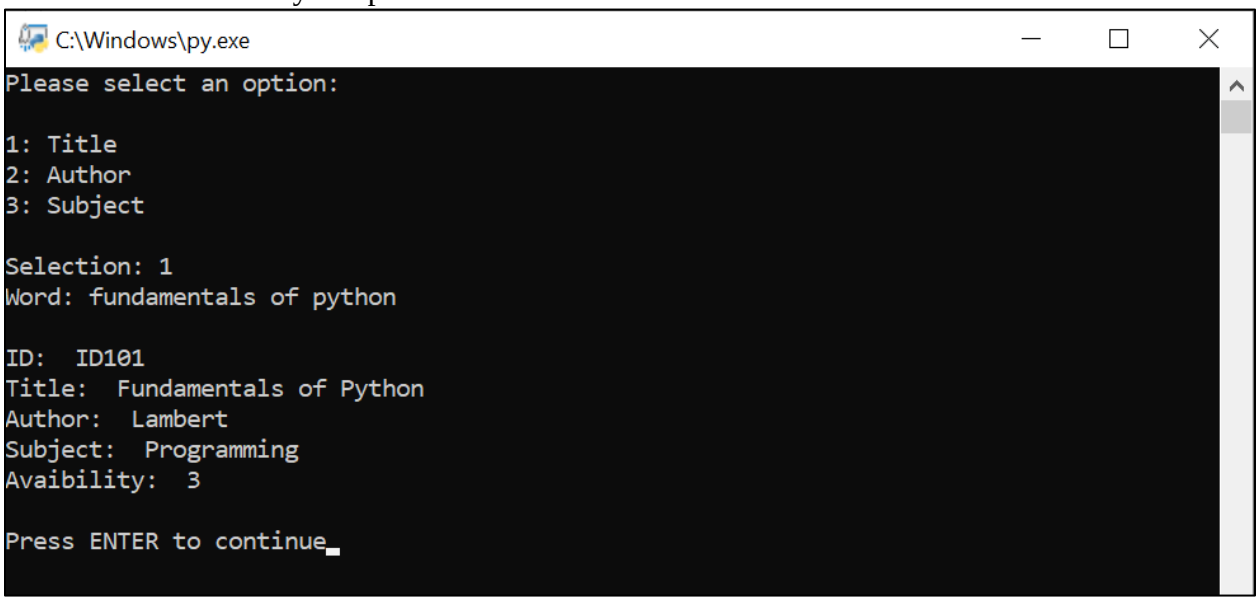
Note:

- The following screen shots show some sample interfaces of different functions or modules.
- It is not compulsory to follow the design of the following interfaces. You are encouraged to design your own interfaces.

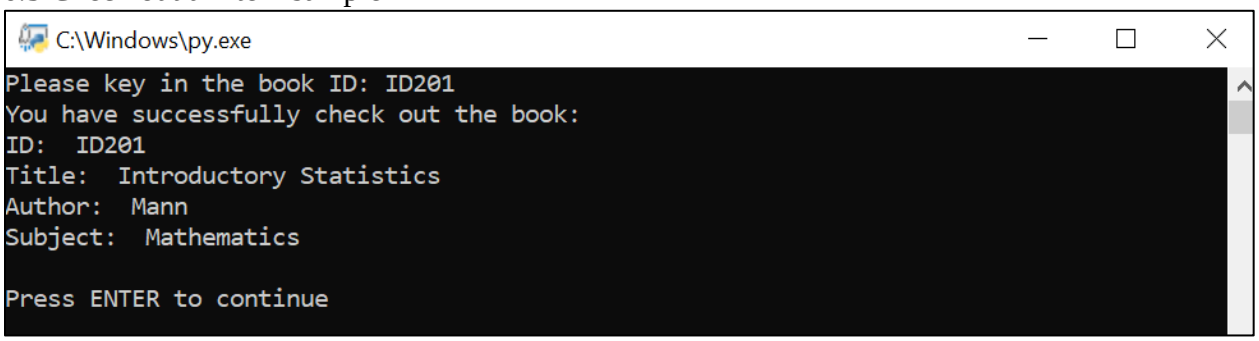
6.1 Main Menu Sample



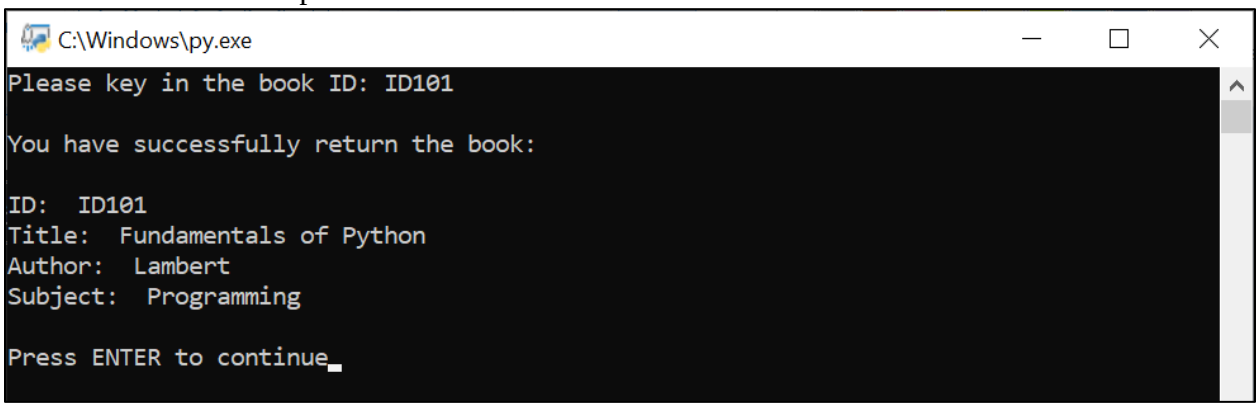
6.2 Search functionality sample



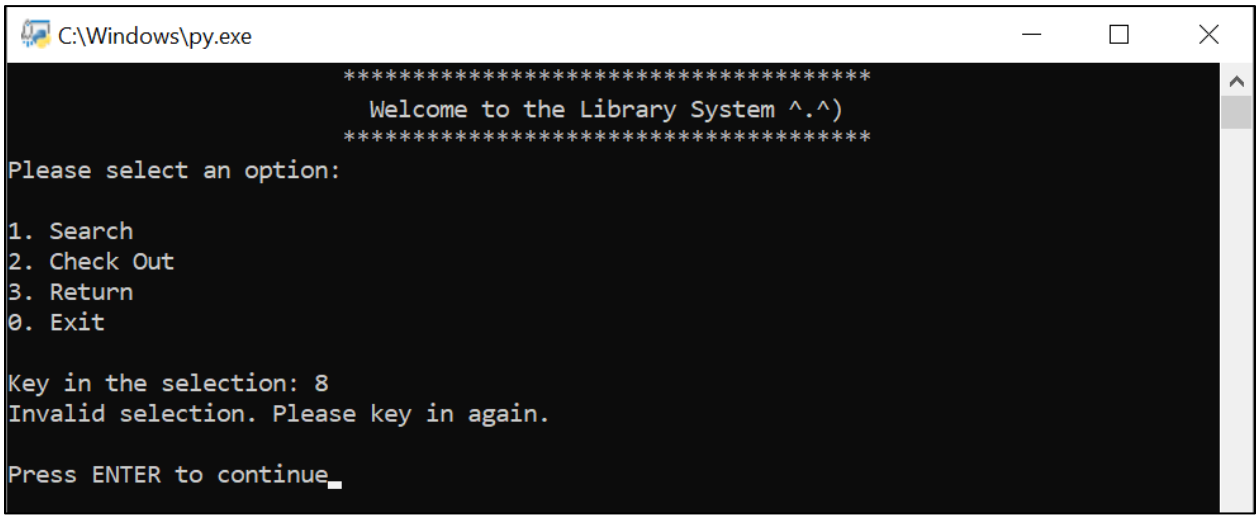
6.3 Check out an item sample



6.4 Return an item sample



6.5 Data validation sample



~ END OF GUIDELINES ~