UNIVERSITY OF Southampton MALAYSIA

COMP1202 - Programming 1
Coursework 1

Module Code:	COMP1202		
Module Title:	Programming 1		
Module Leader:	Abobakr Khalil Al-Shamiri		
Assessment Type:	Individual Coursework	**Weighting:**	25%
Submission Due Date:	27/11/2023 04.00 PM (Malaysia time)		
Method of Submission:	Blackboard submission link is named "Coursework 1 Submission".		

This assessment relates to the following Module Learning Outcomes:

A. Knowledge and Understanding	A1. Simple object-oriented terminology, including classes, objects, inheritance and methods. A2. Basic programming constructs including sequence, selection and iteration, the use of identifiers, variables and expressions, and a range of data types. A3. Good programming style.
B. Subject Specific Intellectual and Research Skills	B1. Analyse a problem in a systematic manner and model in an object-oriented approach.
D. Subject Specific Practical Skills	D1. Design a short program, compile the program, debug the program and test the program. D2. Use simple programming environments to aid the above process.

Coursework Brief:

This coursework 1 mainly consists of **variable**, **method**, **constructor**, **loop**, **selection**, **class**, **object**, **Inheritance**, **ArrayList**, **String** concepts.

This coursework integrates elements from the provided assignments and ensures a logical flow, starting with the creation of the `Library` class and gradually adding item-specific functions, member management, and interactive menu systems.

Submission:

Put all your codes (only .java files) in a single zip folder and submit it to the blackboard submission link by the specified deadline.

Feedback: 2-3 weeks after deadline.

General notes:

- Keep your code clean, less redundant, complete, and easy to understand:
  - Clean: No extra unnecessary codes
  - Less redundant: Don't write the same code twice. Instead, put the useful chunk of codes into methods with proper parameters.
  - Complete: Achieve the basic requirements stated in the question file, but feel free to add more!
  - Easy to understand: Put the proper naming for variables and methods according to the Java convention.
- You can show your distinctive skills by experimenting with new and different things!
- Don't forget to test your codes in the main() method
- Contact us via email for clarifications or any questions!

Rules:

- No work can be accepted after feedback has been given.
- You should expect to spend up to **30 hours** on this assignment.
- Please note the University regulations regarding academic integrity.
- Copying works from/to other students will be penalized under University of Southampton's rule.

Library Management System

Part 1: Library Class [15% marks]

Create a `Library` class that can store items (books, DVDs, magazines) in a library inventory using an appropriate data structure. The `Library` has the following attribute:

- `ArrayList<Item> inventory`: A list of items currently in the library.

Implement the following functions:

1. `addItem(Item item)` [3% Marks]: Adds an item to the library's inventory.
2. `removeItem(String title)` [3% Marks]: Removes an item from the library's inventory by title.
3. `listAvailableItems()` [3% Marks]: Lists all available items in the library's inventory.
4. `listCheckedOutItems()` [3% Marks]: Lists all checked-out items in the library's inventory.
5. `searchItemsByTitle(String title)` [3% Marks]: Searches for items by a specific title and lists them.

Part 2: Item Hierarchy [10% marks]

Create a hierarchy of classes for managing items in a library inventory. The hierarchy should include the following classes:

1. `Item` [4% Marks]: The most general class for all types of items.
2. `Book` [2% Marks]: Represents books in the library inventory.
3. `DVD` [2% Marks]: Represents DVDs in the library inventory.
4. `Magazine` [2% Marks]: Represents magazines in the library inventory.

The `Item` has the following attributes:

- `String title`: The title of the item
- `String year`: The year of the item.
- `boolean checkedOut`: true indicates that this item is checked out, false otherwise.

Ensure proper inheritance relationships between these classes. Implement constructors and getter/setter methods as needed.

Part 3: Book Functions [5% marks]

Implement the following functions specifically in the `Book` class to manage book-specific attributes:

1. `input()` [3% Marks]: Prompts the user to input data for the book, including title, author, year, and any additional book-specific attributes (e.g., ISBN, genre, number of pages).
2. `print()` [1% Mark]: Prints all the book's data, including the title, author, year, and any additional book-specific attributes.
3. `getBookDetails()` [1% Mark]: Returns a formatted string containing all the book's details (title, author, year, and any additional attributes). This function is useful for displaying book information in various contexts.

Part 4: DVD Functions [5% marks]

Implement similar functions as in Part 3, but specific to DVDs, in the `DVD` class:

1. `input()` [3% Marks]: Prompts the user to input data for the DVD, including title, director, release year, and any additional DVD-specific attributes (e.g., runtime, genre).
2. `print()` [1% Mark]: Prints all the DVD's data, including the title, director, release year, and any additional DVD-specific attributes.
3. `getDVDDetails()` [1% Mark]: Returns a formatted string containing all the DVD's details (title, director, release year, and any additional attributes).

Part 5: Magazine Functions [5% marks]

Implement similar functions as in Part 3, but specific to magazines, in the `Magazine` class:

1. `input()` [3% Marks]: Prompts the user to input data for the magazine, including title, publisher, publication year, and any additional magazine-specific attributes (e.g., issue number, topic).
2. `print()` [1% Mark]: Prints all the magazine's data, including the title, publisher, publication year, and any additional magazine-specific attributes.
3. `getMagazineDetails()` [1% Mark]: Returns a formatted string containing all the magazine's details (title, publisher, publication year, and any additional attributes).

Part 6: Library Member Class [15% marks]

Create a `LibraryMember` class to represent library members. Each member has the following attributes:

- `String memberId`: A unique identifier for the member.
- `String name`: The name of the member.
- `ArrayList<Item> checkedOutItems`: A list of items currently checked out by the member.

Each library member will also have the following methods. Choose the proper method return type and parameters.

Implement the following methods in the `LibraryMember` class:

1. `checkOutItem(Item item)` [5% Marks]: Allows a member to check out an item.
2. `returnItem(Item item)` [5% Marks]: Allows a member to return an item.
3. `listCheckedOutItems()` [5% Marks]: Lists all items currently checked out by the member.

Part 7: Execution [30% marks]

Create a `LibraryManagementSystem` class that simulates interactions between library members and the library. Implement an interactive menu system for the library, allowing library members to perform selected functions.

In the `main()` method, simulate the Library Management System with an interactive menu, providing the following options:

1. Register as a New Library Member [4% Marks]
2. Check Out an Item [4% Marks]
3. Return an Item [4% Marks]
4. List Available Items [4% Marks]
5. List Checked-Out Items [6% Marks]
6. Search for Items by Title [6% Marks]
7. Exit the Library Management System [2% Marks]

Notes:

- Ensure that the program keeps running until the user chooses to exit.
- Implement proper error handling and validation for user inputs.
- Provide clear instructions and prompts for each menu option to guide the user.

- Maintain the library's inventory and library members' checked-out items list appropriately based on user actions.

Additional Features [15% marks]:

To demonstrate your creativity, you can add some additional features to the Library Management System, such as:

- Implementing due dates for checked-out items.
- Limiting the number of items a member can check out.
- Implementing a fine system for overdue items.
- Maintaining a history of checked-out items for each member.

Output Example

```
Welcome to the Library Management System!

1. Register as a New Library Member
2. Check Out an Item
3. Return an Item
4. List Available Items
5. List Checked-Out Items
6. Search for Items by Title
7. Exit

Please select an option: 1
Enter your name: John Doe
You are now registered as a library member with ID: 56333

1. Register as a New Library Member
2. Check Out an Item
3. Return an Item
4. List Available Items
5. List Checked-Out Items
6. Search for Items by Title
7. Exit

Please select an option: 4
Available Items:
1. Book - "The Catcher in the Rye" by J.D. Salinger
2. DVD - "Inception" directed by Christopher Nolan
3. Magazine - "National Geographic" (Issue #123)

1. Register as a New Library Member
2. Check Out an Item
3. Return an Item
4. List Available Items
5. List Checked-Out Items
6. Search for Items by Title
7. Exit
```

```
Please select an option: 2
Enter your member ID: 56333
Available Items:
1. Book - "The Catcher in the Rye" by J.D. Salinger
2. DVD - "Inception" directed by Christopher Nolan
3. Magazine - "National Geographic" (Issue #123)
Enter the title of the item you want to check out: The Catcher in the Rye
John Doe checked out: The Catcher in the Rye
Item checked out successfully.

1. Register as a New Library Member
2. Check Out an Item
3. Return an Item
4. List Available Items
5. List Checked-Out Items
6. Search for Items by Title
7. Exit

Please select an option: 5
Enter your member ID: 56333
Checked-Out Items for John Doe:
The Catcher in the Rye

1. Register as a New Library Member
2. Check Out an Item
3. Return an Item
4. List Available Items
5. List Checked-Out Items
6. Search for Items by Title
7. Exit

Please select an option: 3
Enter your member ID: 56333
Checked-Out Items for John Doe:
The Catcher in the Rye
Enter the title of the item you want to return: The Catcher in the Rye
John Doe returned: The Catcher in the Rye

1. Register as a New Library Member
2. Check Out an Item
3. Return an Item
4. List Available Items
5. List Checked-Out Items
6. Search for Items by Title
7. Exit

Please select an option: 7
Thank you for using the Library Management System. Goodbye!
```

------ ***End of file*** ------

Any work submitted after the deadline's time will be subject to the standard University late penalties unless an extension has been granted, in writing by the Senior Tutor, in advance of the deadline. Details on the University's late penalties can be found here:

- https://www.southampton.ac.uk/~assets/doc/quality-handbook/Late%20Submission.pdf