DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

CS994 OBJECT ORIENTED PROGRAMMING 2024/25
INDIVIDUAL COURSEWORK ASSIGNMENT

A few important points:

- *Below is the set of compulsory requirements, i.e. you must complete these correctly in order to qualify for full marks.*
- *The final version of your library system **must use inheritance**. Your system is expected to have: **at least one superclass (parent) with at least two subclasses (children)**. It is your responsibility to specify the appropriate inheritance hierarchy/relationships. Please also check the detailed marking scheme regarding inheritance.*
- *Classes: you are **NOT** allowed to implement additional classes unless this is required to meet the inheritance requirements.*
- *Data fields: in some cases, I have provided only a sub-set of data fields per class. You need more to qualify for full marks. For example, a library member will most likely have a name data field. Use your judgement. In other cases, the data fields are implied in the description of the requirements and/or by a question.*
- *Constructors: ALL classes must include constructors (at least one per class!), which initialise all fields to appropriate values.*
- *Functionality and methods: the requirements are strict; unless I have made a mistake (e.g. wrong parameters, return types etc.), in which case I am happy to acknowledge it and revise the requirements. Ask me if in doubt.*
- *Error checking: the requirements contain some guidance for error checking. In other cases, error checking is implied by asking you a question.*
- *Error messages: appropriate error messages should be printed in all cases of error checking.*
- *For all methods that have objects as parameters, `null` values should be checked, and appropriate messages should be printed on the screen.*

Final requirements: Library system

- A class that represents a physical book, i.e. a resource that can be borrowed, returned etc.
    - *Data fields:* the ISBN, the title, an author object (*refer to the respective class below*), a library member object, i.e. a typical library user (*refer to the respective class below*), and one list of `String` objects that stores the damages for the book. Plus, any other data fields you feel are required.

  - *Functionality:*
    - All getter & setter methods. The setter method for the damages list should append the value of its parameter to the end of the list storing the damages.
    - A method that checks if a book is available: returns `true` if the value of the library member data field is `null`. Otherwise, it returns `false`.
    - A method that prints all the details of a book. What if the value of the library member data field is `null`? What if the value of the author data field is `null`?

- A class that represents an electronic resource, e.g. e-book, online journal article etc.
  - *Data fields:* a `boolean` field that indicates whether the resource can be downloaded. Plus, any other data fields you feel are required.
  - *Functionality:*
    - All getter & setter methods.
    - A method that prints all the details of an electronic resource.

- A class that represents the author of a resource.
  - *Data fields:* one field that holds the author's first name, one field that holds the author's surname, and one field that holds the author's address.
  - *Functionality:*
    - All getter & setter methods.
    - A method that prints all the details of an author.

- A class that represents a library member, i.e. a typical user of a library that can borrow/return books and/or access electronic resources.
  - *Data fields:* one field that holds the library member id number, a list that stores the book objects currently borrowed by a library member. Plus, any other data fields you feel are required.
  - *Functionality:*
    - All getter & setter methods, if applicable.
    - A method that prints all the details of a library member.
    - A method that accepts a book object as a parameter and appends this object to the end of the list that stores the books currently borrowed by a library member. What if this book object is already in the list?
    - A method that prints the details of all books currently borrowed by a library member.
    - A method that returns the number of books currently borrowed by a library member.

- A class that represents a library guest/visitor, i.e. a non-typical user of a library that cannot borrow/return books, but can access electronic resources while at the library.
    - *Data fields:* one integer that represents the duration (number of days) of guest/visitor library access. This cannot be greater than 3 days! Plus, any other data fields you feel are required.
    - *Functionality:*
        - All getter & setter methods, if applicable.

- A class that represents the library.
    - *Data fields:* a **single** list that stores library resources, i.e. the library catalogue. Based on the above, a library resource can be a physical book or an electronic resource. Plus, any other data fields you feel are required.
    - *Functionality:*
        - All getter & setter methods, if applicable.
        - A method that prints all the details of the library.
        - A method that checks if the catalogue contains a resource object. The method accepts one parameter, i.e. the resource object. It returns `true` if the resource is contained in the catalogue. Otherwise, it returns `false`.
        - A method that simulates editing the author's first name for a resource. The method accepts two parameters: the resource object to edit, and the new first name for the author. What if the resource object is not in the catalogue? What if the author data field for a resource is `null`?
        - A method that searches the catalogue by resource title. The method accepts one parameter, i.e. the resource title to search, and prints the details of all resource objects with this title. At the end, it also prints the total number of resource objects found. The search should ignore case, i.e. uppercase/lowercase.
        - A method that searches the catalogue by author surname. The method accepts one parameter, i.e. the author's surname to search, and prints the details of all resource objects by this author. At the end, it also prints the total number of resource objects found. What if the author data field for a resource is `null`?
        - A method that removes a resource object from the library catalogue. The method accepts one parameter, i.e. the resource object to remove. What if this resource is not in the catalogue? What if this resource is currently on loan?
        - A method that removes a resource object at a specific position in the library catalogue. Similar to the previous method, but the parameter represents a position in the catalogue. What if the specified position does not exist in the catalogue? What if this resource is currently on loan?

- A method that prints the details of all available books, i.e. books not currently on loan. What if a resource object in the library catalogue is an electronic resource, i.e. not a physical book?
- A method that returns the number of resource objects in the library catalogue.
- A method that adds a resource object to the catalogue. The method accepts one object as a parameter and appends this object to the end of the library catalogue. If the object already exists in the catalogue, the method should reject it and print an appropriate error message.
- A method that takes one physical book object and one library member object as parameters, and simulates the process of lending a book: the library member data field of the book parameter is set to the value of the library member parameter, and the book parameter is added to the list of currently borrowed books for the library member parameter. These operations cannot be performed in the following cases:
    - The book object is not in the catalogue.
    - The book object is currently borrowed by another library member.

  Appropriate error messages should be printed in all above cases!
- A method that simulates the process of accepting a physical book return. The method takes three parameters: the book object to return, one `boolean` that indicates whether a damage is to be recorded, and the damage description as a `String`. The library member data field of the book parameter is set to `null`, and if a damage is to be recorded, the damages data field of the book is updated accordingly. The operations cannot be performed if the book object is not in the catalogue, in which case an appropriate error message should be printed.
- A method that prints the details of all physical book objects in the catalogue.
- A method that prints the details of all electronic resource objects in the catalogue.

Good luck!