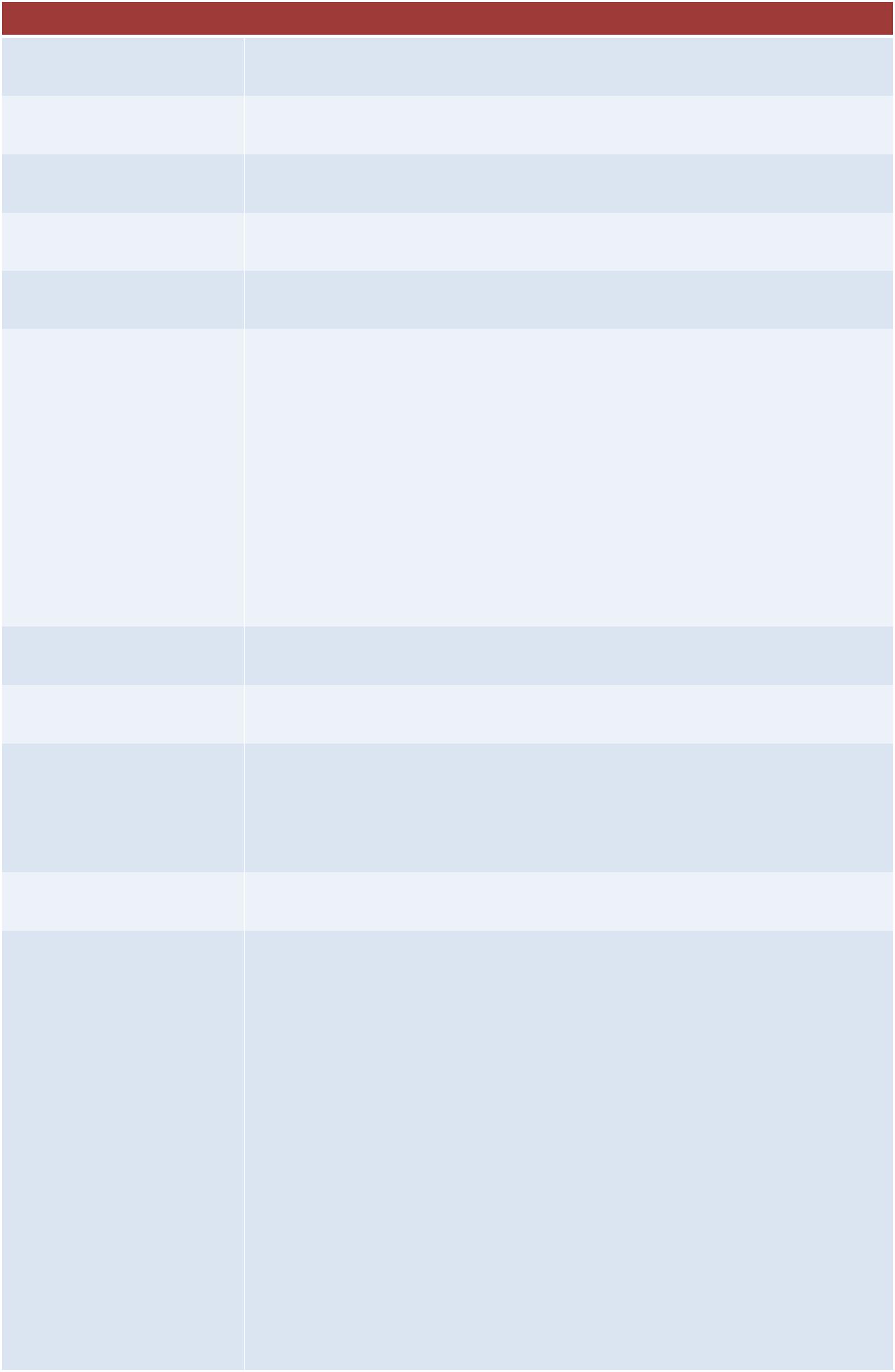
**University of Westminster**

School of Computer Science & Engineering



**5COSC001W** **Object Oriented Programming – Coursework 1 (2018/19)**

Module leader

Unit

Weighting:

Qualifying mark

Description

Learning Outcomes Covered in this Assignment:

Guhanathan Poravi

Coursework 1

50%

30%

Object Oriented Programming and Design

This assignment contributes towards the following Learning Outcomes (LOs):

* LO1 Identify and justify good practices in the development of object oriented software;
* LO2 Apply acquired knowledge of concepts, characteristics, tools and environments to adapt to new computational environments and programming languages which are based on object oriented principles;
* LO3 Design, implement efficiently applications based on a OOP language, given a set of functional requirements.
* LO4 Implement GUI interfaces using an OOP language;

|  |  |  |
| --- | --- | --- |
| Handed Out: | 10th October 2018 | |
| Due Date | Sunday 2nd December 2018 Submissions by 13:00 | |
| Expected deliverables | Submit on Blackboard a zip file containing: | |
|  | **A folder with all the UML documents attached** | |
|  | **A folder with the developed project**  **A report containing code for each functionality and requirements** | |
| Method of Submission: | Electronic submission on BB via a provided link close to the submission time. | |
|  |  |
| Type of Feedback and Due | Written feedback by 24/12/2018 and generic feedback during the | |
| Date: | demonstration. | |
|  | **2.1.1 Knowledge and understanding of facts, concepts, principles &** | |
| BCS CRITERIA MEETING IN | **theories** | |
| THIS ASSIGNMENT | **2.1.2** | **Use of such knowledge in modelling and design** |
|  | **2.1.3** | **Problem solving strategies** |
|  | **2.2.1** | **Specify, design or construct computer-based systems** |
|  | **2.2.4** | **Deploy tools effectively** |
|  | **2.3.2** | **Development of general transferable skills** |
|  | **3.1.1** | **Deploy systems to meet business goals** |
|  | **4.1.1** | **Knowledge and understanding of scientific and engineering** |
|  | **principles** | |
|  | **4.1.3** | **Knowledge and understanding of computational modelling** |

**Assessment regulations**

Refer to section 4 of the “How you study” guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

**Penalty for Late Submission**

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

**Coursework Description**

**Objective**:

The aim of this assessment is to assess the skills and learning that you have acquired about object- oriented programming during the module. You are asked to implement a program in which objects interact in order to fulfill with a set of functional requirements.

**Analyze the statement**:

An important skill that you will start to develop in this module is analyzing a problem statement in order to identify the details needed to develop a solution.

In this assignment, the first task you should perform is a careful analysis of the problem statement in order to make sure you have all the information to elaborate a solution. **Do not make assumptions about what is needed! If you are not sure, about the information provided, please ask the module leader in lecture or your tutor during tutorials**

**Design a solution**:

The design of your system should be consistent with the Object Oriented principles and easy to understand by an independent programmer*.*

You are required to design your program using UML diagrams. In particular you have to draw:

* a class diagram *(5 points)*
* two or more use cases for the system *(5 points)*.

**Problem description and requirement statement**

**You are required to develop a program that implement a Library Management System where the frontend must be developed using type script and any other frameworks on top of it and backend developed using play framework.**

In the library, there is space to store only 100 books and 50 DVDs (150 items in total).

In this assignment, you will be required to create necessary classes to implement the following functionality*:*

1. Design and implement a class **LibraryItem** (abstract) *(3 points)* and the subclasses **Book** and **DVD**. The classes should hold information about the ISBN, the Title, the sector, the Publication Date, the Date/Time it has been borrowed and the current reader (in case the book has been borrowed), and include appropriate methods.

In particular:

* + The **Book** class should include appropriate methods and hold information about the authors (note that it could be also more than one author), the publisher and the total number of pages *(3 points)*.
  + The **DVD** class should include methods and information about the available languages, the available subtitles, the producer and the actors *(3 points)*.
  + The class **Reader** should have methods and information about the ID, Name, mobile number and email *(3 points)*.
  + You should implement a class **DateTime** to represent the time/date of when the item has been borrowed. Do not use any predefined library for date and time and you can refer as example to the class that has been provided during the tutorials *(2 points)*.

1. Design and implement a class called **WestminsterLibraryManager**, which implements the interface **LibraryManager**. WestminsterLibraryManager maintains the list of the items in the library and providesall the methods for the library manager actor to manage the library.

The system should have necessary **User Interface** with following options (a single menu from which you can activate the actions) that can be used to perform the following management actions from which the manager can select one.

* **Add a new item** in the library and display the number of free spaces (remember that the library canstore max 150 items: 100 for books and 50 for DVD). Display a message in case there are no spaces available. The user can select if they would like to add a book or a DVD and enter the corresponding information *(6 points)*.
* **Delete an item**, given the ISBN, from the library and display the number of free spaces left. Displaythe type of the item that has been deleted (if it is a book or a DVD) *(6 points)*.
* **Display the list of the items** in the library. For each item display the ISBN, the type of item (if it is a bookor a DVD) and the title *(3 points)*.
* **Borrow an item** in the library given his ISBN. Consider that you need to save the name of theReader (the person who borrows the item) and the date/time that the reader borrowed it. Also, each book can be borrowed for 7 days maximum and each DVD for 3 days maximum. In cases where the item is currently borrowed by another reader, display a message saying it is not available at the moment and when it will be available again *(11 points)*.
* **Return an item** that has been borrowed. Modify the relative current information in the item thathas been returned and that now can be available again. If the item has been returned after the period allowed, then a fee will be applied: the reader will pay £0.2 any extra hour for the first 3 days, then £0.5 per hour. A message will say how much the reader has to pay to the library if the item is returned late *(8 points)*.
* **Generate a report** with all the items overdue and the corresponding fee. Order the list from theitem that has been borrowed for the longest period *(8 points)*.

When **create a User Interface** from where it is possible to see all the information for eachitem. The UI can be opened selecting an option from the menu console. In your UI application you should be able to resize your application keeping the component in the same “relative” position (Material Design) *(6 points)*.

Regarding the UI, note that all code must be manually written, **do not use any tools that** **generates code automatically**. You are NOT allowed to use design tools with drag and dropfunctionality to create the graphical user interface for any part of this coursework!

You must follow the specification below for the implementation of the UI and the program.

* + 1. You need to show the list of items in the library. You can use a Table to visualise them *(4* *points)*.
    2. The user can search a specific item by title *(5 points)*.
    3. For each item display a green or red flag if it is currently available or borrowed respectively *(5* *points)*.

1. Some challenges:
   * A user should be able to make a reservation if a book is currently not available. Add in this option in to the system. The system should calculate an estimate of when the book will be available and display it to the user. In order to attempt this task, notice that you might change the previous implementation: you need to have saved the history (when it has been borrowed and for how many hours) for each item in order to compute how long it has been borrowed on average. Consider also that more than one user can make a reservation and so the time to wait for the item will be summed up! *(8 points) Write in comments how you ended up with your solution!*
   * You can add some extra features! *(6 points)*.

**Demonstration**

You will be asked to demonstrate the system. The following will be evaluated*:*

* the extent to which the code meets the specification given above
* understanding of the code
* quality of the code (e.g. good use of comments, naming of variables, structure of the code)
* error handling (in particular: if the user insert a wrong parameter, is the code robust to handle it?, are the ID or ISBN unique? Etc.)