University of Westminster
School of Computer Science & Engineering

5COSC019C	Object Oriented Programming – Coursework (2022/23)
Module leader	Saman Hettiarachchi
Unit	Coursework
Weighting:	50%
Qualifying mark	30%
Description	Object Oriented Programming and Design
Learning Outcomes Covered in this Assignment:	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; - LO5 Apply appropriate techniques for evaluation and testing and adapt the performance accordingly
Handed Out:	25th October 2022
Due Date	Friday 6th January 2023 Submissions by 13:00
Expected deliverables	Submit on Blackboard a zip file containing: **1) A folder with all the UML documents, test case plan and report attached** **2) A folder with the developed project (NetBeans Solution with your Java code)** **3) A video where you recorded a demonstration of the functionalities of the implemented system.**
Method of Submission:	Electronic submission on BB via a provided link close to the submission time.
Type of Feedback and Due Date:	Written individual feedback within 15 working days. **All marks will remain provisional until formally agreed by an Assessment Board.**

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

Objectives
The aim of this coursework is to assess the knowledge and skills 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 fulfil with a set of functional requirements.

Analyse the problem statement
An important skill that you will start to develop in this module is analysing 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. If you have any questions please write your queries on the forum on BB.

Problem description and requirement statement
You are required to develop a program that implements a system to manage a **Skin Consultation Centre**.

You should implement a **console system** from where the manager can add new doctors, delete if needed, add or cancel consultations, print and save them as described in detailed below.

You should implement a **Graphical User Interface** (GUI) from where we can see the list of doctors, book or edit consultations for patients, etc. as described below.
For the user interface you are not allowed to use drag and drop tools (such as the Designer in NetBeans), but you can use some external API if you want to add graphs or some more professional components.

In this assignment, you will be required to address the following tasks:

1. Design and classes implementation (Phase 1)
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 UML use case diagram for the system *(6 marks)*.
- A UML class diagram *(6 marks)*

Read carefully the following requirements. It is important that you follow the specifications and your design and implementation must comply with these.
According to the *Inheritance* principle you have to design and implement a super class **Person** and the subclasses **Doctor** and **Patient**.

> The classes **Person** should include appropriate methods in order to comply with the *encapsulation principle* and hold information about the *name*, *surname*, *date of birth* and *mobile number* *(4 marks).* (You can add any other information that you consider appropriate and you can implement additional classes with justification to make the code more robust or user friendly).

In particular:

- The **Doctor** subclass should hold specific information and methods. You should add the *medical licence number* and *the specialisation* *(e.g. cosmetic dermatology, medical dermatology, paediatric dermatology, etc.)* as instance variables and the relative get/set methods *(4 marks)*.

- The **Patient** subclass should hold specific information and methods. You should add a patient *unique id* as instance variables (attribute) and the relative get/set methods *(4 marks)*.

- You should implement a class **Consultation** to represent the booked consultation with a specific doctor from a patient. The class should hold information about: *date and time* slot for the consultation (to represent the date you can use either the class provided during tutorials or you can use any java API), the *cost*, *notes*, and the relative get/set methods *(4 marks)*.

- Design and implement a class called **WestminsterSkinConsultationManager**, which implements the interface **SkinConsultationManager** *(2 marks).* WestminsterSkinConsultationManager maintains the list of the doctors and provides all the methods for the system manager.

2. Console Menu Implementation (Phase 2)

The class **WestminsterSkinConsultationManager** should display **in the console a menu,** containing the following management actions from which the user can select one.

- **Add a new doctor** in the system. It should be possible to add a new doctor, with all the relevant information. You should consider that the centre can allocate a maximum of 10 doctors *(5 marks).*

- **Delete a doctor** from the system, selecting the medical licence number. Display a message with the information of the doctor that has been deleted and the total number of doctors in the centre *(5 marks)*.

- **Print the list of the doctors** in the consultation centre. For each doctor, print on the screen all the stored information. The list should be ordered alphabetically according to the doctor surname *(5 marks)*.

- **Save in a file** all the information entered by the user so far. The next time the application starts it should be able to **read back all the information** saved in the file and continue to use the system (*5 marks*).

3. Graphical User Interface (GUI) Implementation (Phase 3)

Open a Graphical User Interface (GUI) from the menu console.
Note: You can choose how the GUI should look like and how to meet at the best these specifications.

You should implement the GUI according the following requests:

- The user can visualise the list of doctors with relative information. The user should be able to sort the list alphabetically. You are suggested to use a table to display this information on the GUI but you can choose any other solution. *(6 marks)*.

- The user can select a doctor and add a consultation with that specific doctor. When implementing these functionalities, you need to comply with the following requirements:

    - The user can check the availability of the doctor in specific date/time and can book a consultation for a patient if the doctor is available. If the doctor is not available automatically another doctor will be allocated, who is available in that specific date/time. The choice of the doctor has to be done randomly among all available doctors *(6 marks).*

      For each consultation the user has to:

    - Add patient information (add all the attributes defined above - name, surname, date of birth, mobile number, id) (*2 marks*).
    - Enter and save the cost for the consultation. Consider that each consultation is £25 per hour and the first consultation is £15 per hour (*5 marks*).
    - Add some notes (this could be textual information or the user could upload some images of the skin). This information should be encrypted in order to preserver data privacy (*6 marks*). You can use available APIs for the encryption of data.
    - Once the consultation has been saved in the system, the user can select it and visualise all the stored information (*4 marks*).

4. Testing and system validation (Phase 4)

- **Write a test plan** designed to ensure that the coded solution works as expected. The test plan will include specific instructions about the data and conditions the program will be tested with *(5 marks).*

- Implement an automated testing (you can use JUnit or feel free to use any other tool or scripts for unit testing) that runs scenarios of each of the use cases you implemented in the console menu *(4 marks)*.

- The following will be evaluated: ○ The robustness of the code through the use of error handling and input validation *(3 marks).*
    - ○ The quality of the code and the adherence to coding standards and conventions *(3 marks).*

VIDEO – Demonstration (6 marks for the quality of your video)
You are requested to record a video of the system you implemented. You need to run the project and show all the functionalities you implemented in the console menu and in the GUI. Add at least 5 doctors during the demonstration. You should record also your voice explaining what you are demonstrating.
Note that if the video is not submitted your mark will be capped to 30%.
Also, your tutor is entitled to request an in-person viva if there is any doubt about the originality of your work!

Coursework Report
You are request to fill a report about your work. You should download the template from Blackboard and fill the sections with your comments. You can download the template from this link: **Note that if the Report is not submitted your mark will be capped to 30%.**

Coursework Guidance

The implementation goals are split into 4 phases:

Phase 1 This involves an initial design of your system through UML diagrams and the implementation of the classes Person, Doctor, Patient and Consultation (plus any accessory class you might need), and the Interface SkinConsultationManager. The marking scheme allows for students to score up to 30%. I would suggest concluding this Phase 1 by Week 05.

Phase 2 This is for implementing the system functionalities through a console menu. You should provide the implementation (in the class WestminsterSkinConsultationManager) of methods such as add, delete, print, sort doctors and generate report. The marking scheme allows for students to score up to 50%. I would suggest concluding this phase 2 by Week 07.

Phase 3 This phase involves the GUI implementation and the possibility for a user to visualise and book a consulation. The marking scheme allows for students to score up to 85%. I would suggest concluding this phase 3 by Week 09.

Phase 4 This phase is dedicated to the testing and evaluation of your system and the marking scheme allows for students to score up 100% for completing all the phases. This Phase 4 needs to be concluded by week 11 (week of final submission).

Remember to submit the code, the UML diagrams, the test case, the video and report!

Coursework Report – 5COSC019W Object Oriented Programming

To be filled by the student

Student Name:
Student ID:

Have you submitted the video with the demonstration of your system? ☐Yes ☐No

Phase 1 – Design and classes implementation

Task	**Did you attempt the task?**	**Student's comments (To which extent you implemented the task? Have you encountered any problems or issue?)**
Design a UML Use Case Diagram of your system (submitted in a separate file).	☐ Yes ☐ No	
Design a UML Class Diagram of your system (submitted in a separate file).	☐ Yes ☐ No	
Implementation Class Person	☐ Yes ☐ No	
Implementation Class Doctor	☐ Yes ☐ No	
Implementation Class Patient	☐ Yes ☐ No	
Implementation Class Consultation	☐ Yes ☐ No	
Implementation Interface WestminsterSkinConsultationManager	☐ Yes ☐ No	

Phase 2 – Console menu implementation

Task	**Did you attempt the task?**	**Student's comments (To which extent you implemented the task? Have you encountered any problems or issue?)**
Add a doctor in the system with all the relative information (max 10 doctors)	☐ Yes ☐ No	
	☐ Yes ☐ No	

Delete a doctor from the system selecting the medical licence number. Display a message to confirm he/she has been removed and the total number of doctors in the centres.		
Print on the screen the list the doctors in the centre with all the relative information. The list should be ordered alphabetically.	☐ Yes ☐ No	
Save in a file entered by the user so far. The user should be able to load back the information running a new instance of the application.	☐ Yes ☐ No	

Phase 3 – GUI Implementation

Task	**Did you attempt the task?**	**Student's comments (To which extent you implemented the task? Have you encountered any problems or issue?)**
Doctor list visualisation. Sorting alphabetically.	☐ Yes ☐ No	
The user can select a doctor and add a consultation.	☐ Yes ☐ No	
In the consultation the user can add all the patient details.	☐ Yes ☐ No	
The user can select the date/time of the consultation considering that a doctor cannot have more than one consultation at the time.	☐ Yes ☐ No	
The user can enter and save the cost for the consultation. (£25 per hour and only the first one £15).	☐ Yes ☐ No	
The user can add some notes (text information or images). This information has been encrypted.	☐ Yes ☐ No	

Phase 4 – Testing and system validation

Task	Did you attempt the task?	Student's comments (To which extent you implemented the task? Have you encountered any problems or issue?)
Test plan. (Submitted in a separate file).	☐ Yes ☐ No	
Implementation of an automated unit test for each scenario in the console menu.	☐ Yes ☐ No	
Error Handling across all the code, input validation and code quality.	☐ Yes ☐ No	