University of Wolverhampton
Faculty of Science and Engineering
School of Mathematics and Computer Science

Module Assessment

Module	5CS019 Object Oriented Design and Programming
Module Leader	Er Subash Bista
Semester	3
Year	2023
Assessment type	Individual
% of module mark	100%
Due Date	You should aim to complete this work by week 13. You are advised to stop working on this task at that point and move on to the next task. You should speak to your tutors during scheduled workshop sessions to get feedback on your work. The final portfolio submission deadline is **Week 13**.
Hand-in – what?	A Zip file containing program code (**java files**) for the tasks and other specified portfolio items(**UML class diagrams** in png or jpeg image files ,**explanation of UML Class diagram in .pdf file and database files(.sql file) and screenshots of database's table**). Other file formats that cannot easily be opened by your tutor will not be accepted.
Hand-in- where?	To be submitted via Canvas.
Cheating, plagiarism, collusion	Any evidence of these offences will not be tolerated. The work you submit must be your own work.
Pass mark	40% is required overall to pass the module. Each portfolio task has an indicative weighting. However, the final grade will not be calculated entirely mechanically. A degree of academic judgment will be used to assess how well you have met the learning outcomes overall. You must attempt and submit all portfolio tasks.
Method of retrieval	Resit portfolio tasks will be in the summer resit period. These will not necessarily be the same as the original tasks, but variants of them.
Collection of marked work	Feedback returned through Canvas

Important note	To pass this module, students must implement GUI Programming(Swing) and JDBC in this assignment. If not implemented you will directly go for a resit examination.

Task Details

Course Management System

Consider a course registration system for a college. Students are expected to register for the course of study of their choice. Additionally, an administrator at the college does occasionally add new courses to the list of available ones for students to have a wider selection of courses. It is possible for an administrator to cancel a course if it is no longer being offered. A cancelled course might be available at another time. However, when a course becomes unpopular, it is deleted altogether and cannot be reactivated at a later date. A deleted course is permanently removed from the list of courses at the college.

An example course of study is BSc in Computer Science. Courses are delivered on a semester basis, with each semester having 4 modules that students must study, a total of 8 modules altogether to complete a year of study.

An instructor is assigned up to 4 modules across the different levels of study. Students in levels 4 and 5 have no optional modules; however in level 6, students have two electives and 2 mandatory modules. To proceed between levels, a student must pass half of the modules for the current level. A pass mark constitutes a score of at least 40% on assessed work for the module.

Users of the system

Student – a student is a key stakeholder of this system as they choose courses, and enroll for modules on that course. At level 6, the student is also able to select from some optional modules during semesters 1 and 2. Students also are able to view which instructor(s) are teaching on the modules they are studying.

Course administrator – they can add new courses, add modules to a course, cancel a course, or delete a course altogether. They also are responsible for edits/amendments(add) of course names, module names, etc. An administrator is able to generate a report/result slip for a student depicting modules done for the semester or year of study, and the marks and grades for each module. Based on this, there is a decision appended to the result slip indicating whether a student will progress to the next level of study or not.

Instructor (Teacher) – they are assigned modules to teach, can view what modules they are on, and the students registered on those modules. Instructors are able to add marks to each module they are responsible for teaching.

Key features required of the system

- Ability to amend (add) courses and modules.
- Ability to amend(add) instructor details or to remove instructors from modules.
- Ability to produce a result slip for each student indicating their grades on each module
- Persistence of data in database– all data is saved between program invocations

Additionally, there should be evidence of application of object oriented concepts, such as inheritance, encapsulation, object associations and polymorphism.

Your task

UML class diagram

You are to produce a UML class diagram of your solution. You are free to use any tool you like, e.g., MS Visio, StarUML. The construction of the class diagram can be iterative in nature, you can start with the more obvious classes such as Student, Module, Course, and then add to these as you go along.

Your classes in the class diagram should have comprehensive details (class name, attributes, and operations). For operations, you do not need to depict any getter and setter methods as these are rather obviously expected for most attributes. Relationships between classes should be clearly shown.

Application code

You are to implement your design using Java, making sure to exploit object oriented concepts in your implementation. In this task, these concepts **must be used**:

- Encapsulation and data hiding
- Inheritance and polymorphism
- Exception handling (User defined exception and Java Exception class)

Other software development concepts that **must be** used are:

- Data storage (JDBC)
- GUI programming (Swing)

The main areas to pay attention to are:

1. Design
   1.1. UML class diagram – with relevant details about each class
   1.2. Relationships between classes (Inheritance, Associations, Dependency, Multiplicity in class diagram)
   1.3. Matching of class diagram and code (Not completely partially)
2. Code
   2.1. Functionalities (Case study requirements- users functionalities to be implemented)
   2.2. Layout and style (GUI LayoutManager and other GUI components)
   2.3. Use of OOPs concepts (Encapsulation, Inheritance, Association)
   2.4. Use of GUI Programming (Swing)
   2.5. Use of JDBC (Database)
   2.6. Use of Exception handling (User-defined exception and Java Exception class)
3. GUI Output and results (e.g. GUI view of current modules for the course, GUI view of result slip of the Student)