INTI International College Penang **School of Computing**

3+0 Bachelor of Science (Hons) in Computer Science, in collaboration with Coventry University, UK

3+0 Bachelor of Science (Hons) in Computing, in collaboration with Coventry University, UK

Coursework cover sheet

Section A - To be completed by the student.

Full Name: Oon Jay Von	
CU Student ID Number: 14196254	
Semester: 1	
Session: **April 2023**	
Lecturer: **Puteri Nursyawati Azzuri (puteri.azzuri@newinti.edu.my)**	
Module Code and Title: **4067CEM Software Design**	
Assignment No. / Title: **Continuous Assessment**	% of Module Mark: **50**
Hand out Date: **12 May 2023**	Due Date: **Task 1: 02 June 2023, by 11.59pm.** **Task 2: 07 July 2023, by 11.59pm** **Task 3: 23 June 2023, by 11.59pm.** **Task 4: 23 June 2023, by 11.59pm.** **Task 5: 23 June 2023, by 11.59pm.**
Penalties: No late work will be accepted. If you are unable to submit coursework on time due to extenuating circumstances, you may be eligible for an extension. Please consult the lecturer.	
Declaration: I/we the undersigned confirm that I/we have read and agree to abide by the University regulations on plagiarism and cheating and Faculty coursework policies and procedures. I/we confirm that this piece of work is my/our own. I/we consent to the appropriate storage of our work for plagiarism checking. Signature(s): Jay	

Section B - To be completed by the module leader

Intended learning outcomes assessed by this work:

1. Understand and apply appropriate concepts, tools, and techniques to each stage of the software development.

2. Understand and apply design patterns to software components in developing new software.

3. Demonstrate an understanding of project planning and working to agreed deadlines, along with professional, interpersonal skills and effective communication required for software production.

5. Demonstrate an awareness of, and ability to apply, social, professional, legal, and ethical standards as documented in relevant laws and professional codes of conduct such as that of the Malaysian National Computer Confederation.

Marking scheme	Max	Mark
1. User Story Mapping	20	
2. Setting up a GitHub Repository	10	
3. Creating a Class diagram and design pattern selection	30	
4. Creating a Prototype User Interface and Usability Testing	20	
5. Discuss the ethical issue related to the software	20	
Total	100	

Task 3 – Creating a Class diagram and design pattern selection (30 marks)

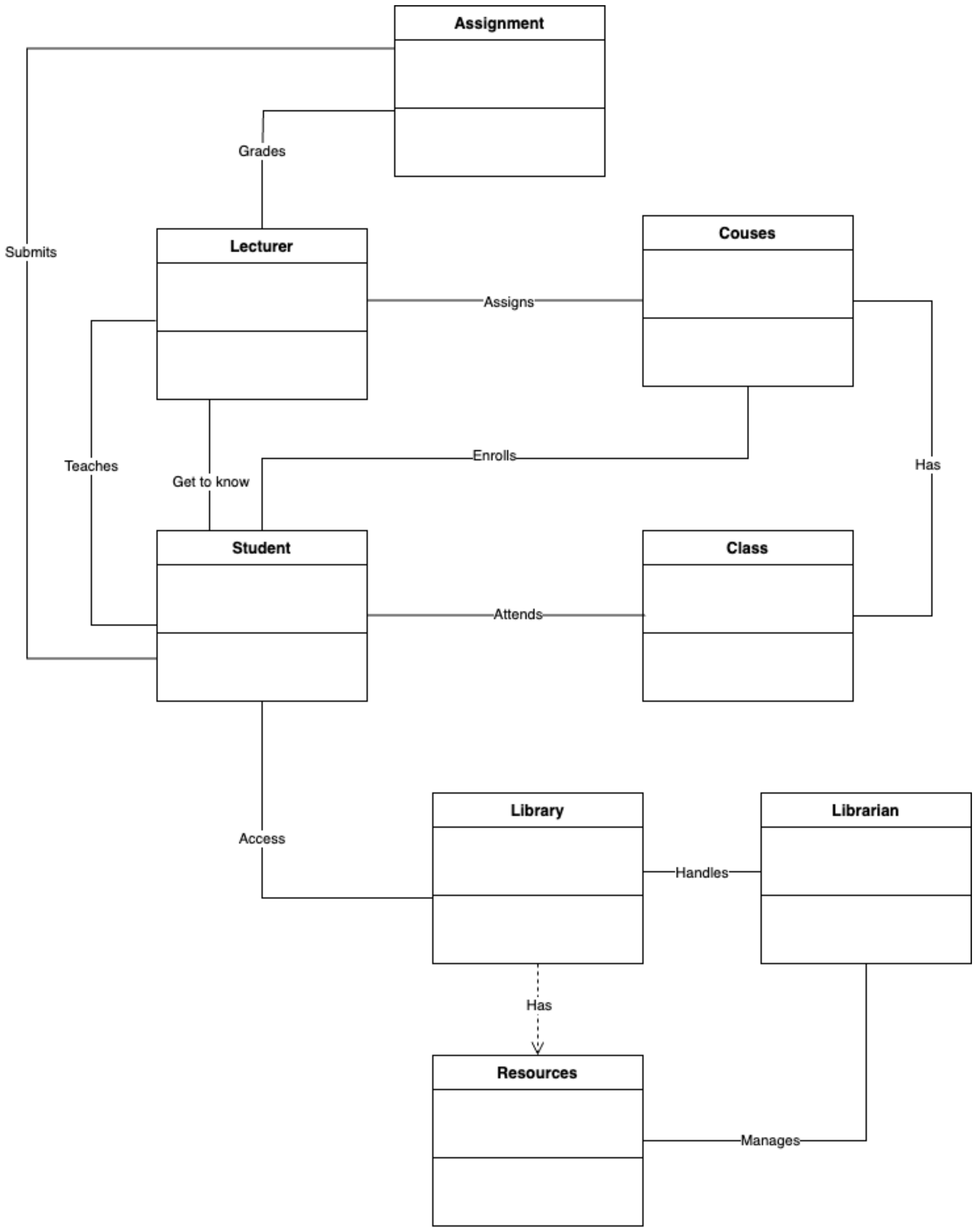
Create a simple Class diagram which should consists of the Classes that might be used to represent the system and the association between them. You don't have to declare the attributes and operations for this activity. You do have to explain the class responsibility of each class declared. You can use software like StarUML to complete this activity.

Output – A class diagram containing classes and associations. In Word format, uploaded it to GitHub.

Consider the problem and select a suitable design pattern that can be implemented on the problem. Give justification on why the design pattern was chosen. Draw the UML diagram representing your class diagram as a design pattern UML. Include all the abstract class/interface, concrete class, and inheritance (if any) used to represent the problem.

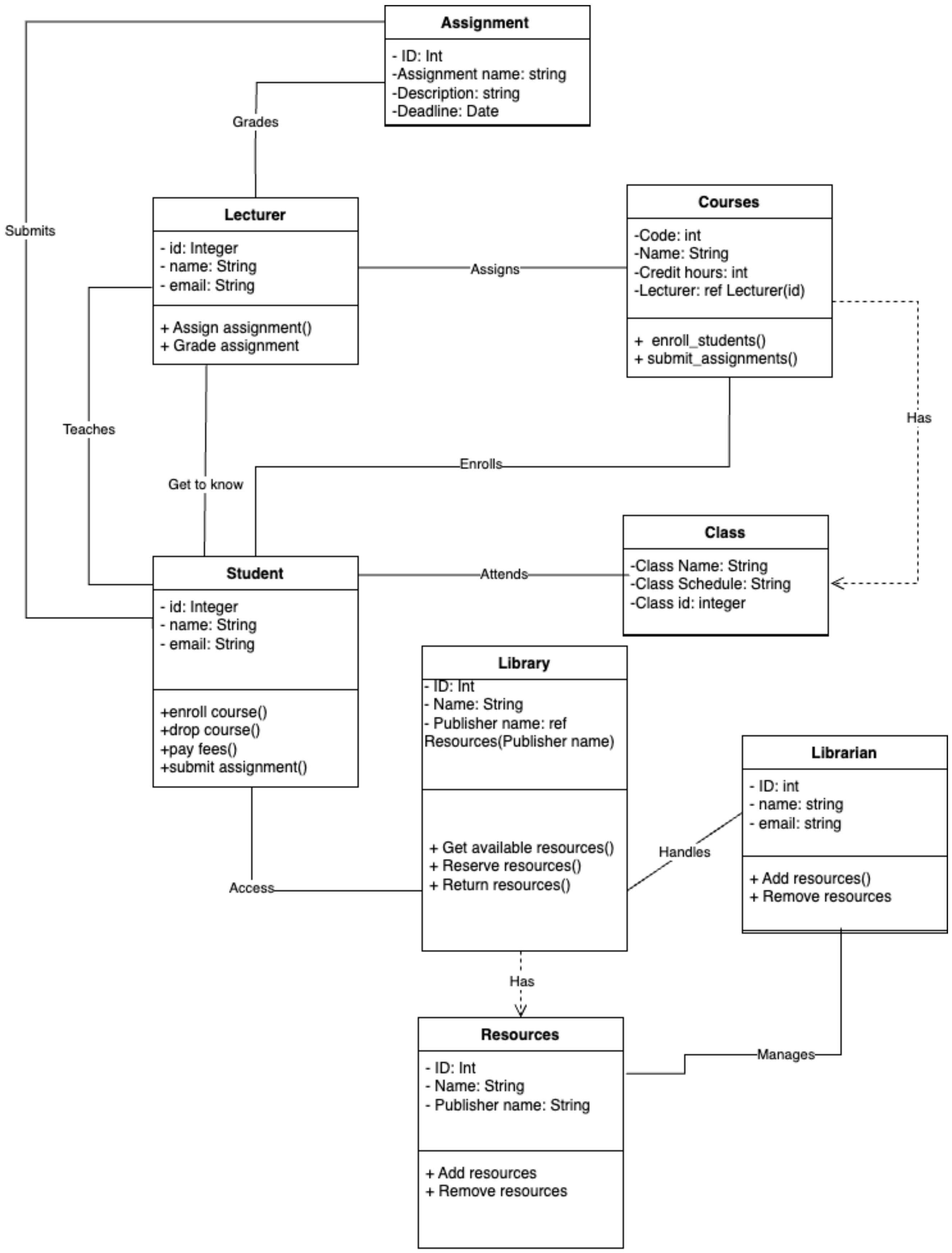
Output – UML diagram representing the design pattern. In Word format, uploaded it to

GitHub. Due – Week 12 of the semester. 23 June 2023, by 11.59 pm.


Assignment
Grades
Submits
Lecturer
Couses
Assigns
Teaches
Get to know
Enrolls
Has
Student
Class
Attends
Access
Library
Librarian
Handles
Has
Resources
Manages

Class responsibilities:

In this class diagram, students will enroll in courses and after enrolling, a lecturer will be assigned to each course, and the courses will have classes for students to attend. Then, the lecturer will be teaching students. Students will submit their assignments and the lecturer will need to grade their assignments. Other than that, students can access the library. In the library, it depends on the resources so that students can get the recommended books or textbooks from the library. Moving on, the library will be managed by the librarian for he/she to take care of the library.


Assignment
- ID: Int
-Assignment name: string
-Description: string
-Deadline: Date
Grades
Submits
Lecturer
- id: Integer
- name: String
- email: String
+ Assign assignment()
+ Grade assignment
Assigns
Courses
-Code: int
-Name: String
-Credit hours: int
-Lecturer: ref Lecturer(id)
+ enroll_students()
+ submit_assignments()
Has
Teaches
Get to know
Enrolls
Class
-Class Name: String
-Class Schedule: String
-Class id: integer
Student
- id: Integer
- name: String
- email: String
+enroll course()
+drop course()
+pay fees()
+submit assignment()
Attends
Library
- ID: Int
- Name: String
- Publisher name: ref Resources(Publisher name)
+ Get available resources()
+ Reserve resources()
+ Return resources()
Librarian
- ID: int
- name: string
- email: string
+ Add resources()
+ Remove resources
Handles
Access
Has
Resources
- ID: Int
- Name: String
- Publisher name: String
+ Add resources
+ Remove resources
Manages

For the above class diagram, each entity has already included their attributes and operations. Lecturer, Student, and Librarian will store their id, name, and email address in integer, string, and string data type. Assignment will store the id in integer type, assignment name in string type, description in string type, and the deadline in date type. Moving on, courses will have the code in integer type, name in string type, credit hours in integer type, and lecturer which is references from the id of the lecturer entity. In the Resources entity, it will store the id in integer type, name in string type, and publisher name in string type whereas the library entity will have the id in integer type, name in string type, and the publisher name which is references from the publisher name of the resources entity.

Other than that, Lecturers will be assigned to courses and will be teaching students as well as grading their assignments. Students will enroll into courses to attend classes and submit their assignments. They can also get to know their lecturers through the student business system. Furthermore, they can access the library to get resources. For the courses, it will have classes which students will be attending. Librarian will handle the library so that it will function normally. Library will be having the resources so that students can get it from the library.

For the student business system for college, I would choose the Creational Pattern known as the Factory Method Pattern. The Factory Method Pattern provides a way to create objects without specifying their exact classes, allowing for flexibility and extensibility in the system. This pattern is particularly suitable for scenarios where the specific type of object to be created may vary or is determined at runtime.

In the context of the student business system for college, there may be different types of student business activities, such as event registration, course enrollment, or fee payment. Each of these activities may have specific requirements and behaviors associated with them. By implementing the Factory Method Pattern, we can encapsulate the creation of these activities into separate factory classes.

The justification for choosing the Factory Method Pattern in this scenario is that it allows us to abstract the creation logic of student business activities from the client code. The client code, which could be the main system or other components, can simply rely on the factory interface to create the desired activity objects without being concerned about the specific implementation details.

Furthermore, the Factory Method Pattern promotes the Open-Closed Principle, which states that software entities (classes, modules, functions) should be open for extension but closed for modification. This means that if we need to introduce new types of student business activities in the future, we can simply create a new factory class without modifying the existing code. This promotes code maintainability and reduces the risk of introducing bugs or breaking existing functionality.

By utilizing the Factory Method Pattern, the student business system can handle the creation of different types of activities in a modular and extensible manner. This design pattern promotes loose coupling and separation of concerns by abstracting the creation logic, making it easier to manage and evolve the system over time. Additionally, it allows for future scalability and flexibility by enabling the addition of new activity types without modifying the existing codebase.

In summary, the Factory Method Pattern is a suitable choice for the student business system for college because it provides a flexible and extensible approach to creating different types of student activities. It encapsulates the creation logic, promotes the Open-Closed Principle, and enables loose coupling between client code and activity objects.

Marking Rubric for Continuous Assessment

	Marks Below 40%	**Marks in the range 40 – 49%**	**Marks in the range 50 – 59%**	**Marks in the range 60 – 69%**	**Marks 70% and above**
User Story Mapping (20 marks)	User Story Mapping not done or User Story copied/does not match the exact system.	User Story Mapping done at a minimum level and does not capture the important activities of the system.	User Story Mapping done and does capture several important activities of the system. The breakdown of the user story mapping can be improved.	User Story Mapping done and does capture several important activities of the system. The breakdown of the user story mapping is good and uses software that can assist that process (For example Trello compared to Ms. Word).	User Story Mapping done and does capture most important activities of the system. The breakdown of the user story mapping is excellent and uses software that can assist that process (For example Trello compared to Ms. Word).
Setting up a GitHub Repository (10 marks)	GitHub repository does not exist or cannot be accessed or the required files are not available at the time of access.	GitHub repository exist and some of the required files are not available at the time of access.	GitHub repository exist and most of the required files are available at the time of access. However the dates does not follow the required deadline.	GitHub repository exist and all of the required files are available at the time of access. However the dates for some files does not follow the required deadline.	GitHub repository exist and all of the required files are available at the time of access. The dates on the files follows the required deadline.
Creating a Class diagram and design pattern selection (30 marks)	The Class diagram does not represent the required solution (contains generic or non-related classes such as admin), the design pattern suggested is not suitable for the given problem.	The Class diagram and design pattern represent the required solution but in a very general and incomplete way. Required classes in the design are not declared.	The Class diagram and design pattern represent the required solution in a partial way. A few required classes in the design are not declared.	The Class diagram and design pattern represent the required solution in a satisfactory way. Most required classes are declared.	The Class diagram and design pattern represent the required solution in an excellent way. All required classes are declared.

Creating a Prototype User Interface and Usability Testing (20 marks)	No prototype were available or the measurement for the usability testing is not clear.	The prototype cover minimalist and trivial design (such as login) and the measurements for the usability testing are not clear.	The prototype cover adequate design and several measurements for the usability testing are not clear.	The prototype cover good design and most measurements for the usability testing are clear.	The prototype cover excellent design and all measurements for the usability testing are clear.
Discuss the ethical issue related to the software (20 marks)	There is no discussion on the ethical issue or only the theories are pasted back for this component.	There is an attempt to discuss on the ethical issue but no critical analysis was done	There is an attempt to discuss on the ethical issue with some critical analysis was done	There is an attempt to discuss on the ethical issue with good critical analysis.	There is an attempt to discuss on the ethical issue with excellent critical analysis.