

**ACADEMIC YEAR 2019/20 SEMESTER 1**

**LEVEL: 5**

**MODULE TITLE: Object Oriented Programming Development**

**MODULE CODE: MOD005437**

**Assessment Element: 010**

# Word Count: 1,000 words equivalent

# Assessment Weighting: 30% Fine Graded

**The deadline by which this assignment must be demonstrated is:**

**2pm on Friday December 13, 2019**

**YOUR TASKS**

You are required to complete a series of Object Oriented Design and Development tasks (1000 words equiv):

**TASK ONE: Class Design – Learning Outcome 1 and 4**

Using UML style Diagrams and appropriate notation, design a solution for the scenario below. You should clearly demonstrate the relationships between classes and the properties and methods of each class included.

* Design a Student class object for a University. Think about what data you would need to provide for a system such as e-vision or ProMonitor. Ensure to read the rest of this brief to ensure you have included all required details.
* The program should include a Grade Profile which stores a list of grades that the learner has achieved.
* Each student should have a Grade Profile
* It should be possible to create a Grade object which represents the individual grade for an assignment on a module.
* The Grade Profile should be able to calculate the student average grade for a year of study. As well as any Class diagrams you should provide pseudocode for this operation.
* Provide as much detail you think would be necessary to create a student report card which displays their grade information for each module and average grade for the year.
* As an extension, can you add a Weight value to the Grade which represents the percentage of the total mark for a module that an assignment represents?

e.g.

Module=Object Oriented Programming,

Assignment = 1,

Mark = 74,

Weight = 0.3

Module=Object Oriented Programming,

Assignment = 2,

Mark = 65,

Weight = 0.7

**At a minimum, you should provide a set of Class Responsibility and Collaborator Cards, and a Class Diagram to accurately represent the above scenario, along with pseudocode to represent any methods that would process the data provided. You should also provide a Sequence Diagram that represents the order of operations and lifecycle of the objects in the system.**

**As an addition, you may wish to provide further UML documentation to represent the final system, such as Activity and State diagrams.**

**TASK TWO: Class implementation – Learning Outcome 3**

Using C# and DotNet Core

For the scenario designed in Task 1, implement the design as accurately as you can using C# with DotNet Core

* Implement the project using C# with DotNet Core.
* Create the Grade Profile class and create an instance of this class related (using either association/aggregation/composition) to the Student class.
* Create the Grade class which represents the individual grade for an assignment on a module.
* Within the Grade Profile class add the method to calculate the student average grade for a year of study.
* Implement an output screen to display the student report card.
* Create instances of two students with example data, add them to a collection and output their grade profiles and average grades to the screen
* As an extension, can you add a Weight value to the Grade class which represents the percentage of the total mark for a module that an assignment represents?

e.g.

Module=Object Oriented Programming,

Assignment = 1,

Mark = 74,

Weight = 0.3

Module=Object Oriented Programming,

Assignment = 2,

Mark = 65,

Weight = 0.7

Consider how this will affect your average grade calculations and modify your program accordingly.

**TASK THREE: SOLID Principles – Learning Outcome 2**

There are 5 key principles known collectively as SOLID that when implemented together help you to produce code which is easy to maintain, extend and refactor. The principles are:

* S – Single-responsibility principle
* O – Open-closed principle
* L – Liskov substitution principle
* I – Interface segregation principle
* D – Dependency Inversion Principle

Modify the C# code sample provided on GitHub (**010 Task 3** link on Canvas) so that it meets each of the SOLID principles, in order to provide the final functionality as outlined below.

The library system should allow the user to add books to the system with the following properties:

ID

Title

Author

Publisher

Date

Published

Category

* There needs to be an option to add fiction books that also contain a Genre
* There needs to be an option to add a book that has multiple Authors
* The user should be able to display a list of all books in the catalogue including both Non- fiction and Fiction together
* Books should be able to be checked out and back into the library.

As an extension

* The system should allow the user to be able to choose to save and load the book data in either JSON or XML formats, it is expected that this could also be extended in the future to save in other formats.
* Allow the user to search for books by title and return the details for matching results.

Any assumptions you make or extensions to the required functionality should be noted in the comments.

You should add comments to any changes you make explaining how the new version improves the code following SOLID principles and which principles have been met.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MARKING** | **Criteria** | **Grade /25** |  | **Grade /25** |  | **Grade /25** |  | **Grade /25** |
| **Class Design** | All classes correctly defined using UML class notation via CRC and Class Diagrams including appropriate collaborators, responsibilities, attributes, properties and methods to meet the brief. |  | Accurate relationships identified and indicated between classes using appropriate UML notation and pseudocode provided for the core Methods. As an extension include the assignment weighting in the calculations. |  | A sequence diagram accurately representing the interaction between objects in the system and their lifecycles, using appropriate UML notation. |  | Further UML diagrams to accurately define the operation of the system designed, such as State and Activity diagrams. |  |
| **Class Implementation** | All classes accurately represent both the defined requirements and the CRC/class diagram |  | Code standards are high and follow a consistent convention. Object Oriented Principles are adhered to. |  | Well designed and implemented user interface with robust interaction (preventing user errors where possible) |  | Program functions correctly and produces the average grades for each student. As an extension this factors in the assessment weightings. |  |
| **SOLID Principles** | The system functions as described in scenario for adding and managing books |  | The user is able to save and load files in either JSON or XML format. The user can also search for books in the collection. |  | Comments accurately identify where a SOLID principle was not being met and how the code was adapted/refactored in response. |  | Final implementation clearly follows all 5 SOLID principles its implementation. |  |

Each of the three tasks are equally weighted and therefore each represent 10% of the 30% available for this element.

**TURNITIN**

You do not need to upload these tasks to Turnitin. The tasks should be uploaded to Canvas by the deadline.

**YOUR FEEDBACK**

**FORMATIVE FEEDBACK**

Throughout the module you are advised to take advantage of formative feedback opportunities. In addition to Turnitin these are:

Week 4 – Assignment 010

Week 7 – Assignment 010

Week 11 – Assignment 011

**SUMMATIVE FEEDBACK**

You will receive comments on your work within 20 working days of the deadline detailed above. This is Friday January 22nd, 2020. Your feedback may include your un-moderated fine grade, but please note that this will remain subject to change until it has been ratified by the Module Approval Panel. Your final, approved grade will appear on your student record by:

1st February for Semester 1 modules

1st July for Semester 2 modules

1st September for resits and all other August deadlines

Your work will be assessed against the following learning outcomes; in undertaking the tasks outlined above, you should ensure that you evidence these. Once your assessor has confirmed that you have met the requirements of each learning outcome, the assessment criteria and marking standards (below) will be used to decide your fine grade.

|  |  |
| --- | --- |
| **Learning Outcomes (threshold standards):** | |
|  | On successful completion of this module the student will be expected to be able to: |
| **Knowledge and understanding** | 1. Identify and select UML tools for Object Oriented development  2. Apply SOLID techniques to the design of Object Oriented programs |
| **Intellectual, practical, affective and transferable skills** | 3. Develop Object Oriented solutions to real world programming problems  4. Create appropriate documentation for OO solutions |

**LEVEL 5 Assessment criteria and marking standards**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Level 5** reflects continuing development from Level 4. At this level students are not fully autonomous but are able to take responsibility for their own learning with some direction. Students are expected to locate an increasingly detailed theoretical knowledge of the discipline within a more general intellectual context, and to demonstrate this through forms of expression which go beyond the merely descriptive or imitative. Students are expected to demonstrate analytical competence in terms both of problem identification and resolution, and to develop their skill sets as required. | | | | |
| **Mark Bands** | | **Outcome** | **Generic Learning Outcomes (GLOs) (Academic Regulations, Section 2)** | |
| **Knowledge & Understanding** | **Intellectual (thinking), Practical, Affective and Transferable Skills** |
| **Characteristics of Student Achievement by Marking Band** | 90-100% | *Achieves module outcome(s) related to GLO at this level* | Exceptional information base exploring and analysing the discipline, its theory and ethical issues with extraordinary originality and autonomy. | Exceptional management of learning resources, with a higher degree of autonomy/ exploration that clearly exceeds the brief. Exceptional structure/accurate expression. Demonstrates exceptional intellectual originality and imagination. Exceptional team/practical/ professional skills. |
| 80-89% | Outstanding information base exploring and analysing the discipline, its theory and ethical issues with clear originality and autonomy | Outstanding management of learning resources, with a degree of autonomy/exploration that clearly exceeds the brief. An exemplar of structured/accurate expression.  Demonstrates outstanding intellectual originality and imagination. Outstanding team/practical/professional skills |
| 70-79% | Excellent knowledge base, exploring and analysing the discipline, its theory and ethical issues with considerable originality and autonomy | Excellent management of learning resources, with a degree of autonomy/exploration that may exceed the brief. Structured/accurate expression. Excellent academic/ intellectual skills and team/practical/professional skills |
| 60-69% | Good knowledge base; explores and analyses the discipline, its theory and ethical issues with some originality, detail and autonomy | Good management of learning with consistent self- direction. Structured and mainly accurate expression. Good academic/intellectual skills and team/practical/ professional skills |
| 50-59% | Satisfactory knowledge base that begins to explore and analyse the theory and ethical issues of the discipline | Satisfactory use of learning resources. Acceptable structure/accuracy in expression. Acceptable level of academic/intellectual skills, going beyond description at times. Satisfactory team/practical/professional skills.  Inconsistent self-direction |
| 40-49% | *A marginal pass in module outcome(s) related to GLO at this level* | Basic knowledge base with some omissions and/or lack of theory of discipline and its ethical dimension | Basic use of learning resources with little self-direction. Some input to team work. Some difficulties with academic/ intellectual skills. Largely imitative and descriptive. Some difficulty with structure and accuracy in expression, but developing practical/professional skills |
| 30-39% | *A marginal fail in module outcome(s) related to GLO at this level. Possible compensation. Sat- isfies qualifying mark* | Limited knowledge base; limited understanding of discipline and its ethical dimension | Limited use of learning resources, working towards self- direction. General difficulty with structure and accuracy in expression. Weak academic/intellectual skills. Still mainly imitative and descriptive. Team/practical/professional skills that are not yet secure |
| 20-29% | *Fails to achieve module outcome(s) related to this GLO. Qualifying mark not satisfied. No compensation available* | Little evidence of an information base. Little evidence of understanding of discipline and its ethical dimension | Little evidence of use of learning resources. No self- direction, with little evidence of contribution to team work. Very weak academic/intellectual skills and significant difficulties with structure/expression. Very imitative and descriptive. Little evidence of practical/professional skills |
| 10-19% | Inadequate information base. Inadequate understanding of discipline and its ethical dimension | Inadequate use of learning resources. No attempt at self- direction with inadequate contribution to team work. Very weak academic/intellectual skills and major difficulty with structure/expression. Wholly imitative and descriptive.  Inadequate practical/professional skills |
| 1-9% | No evidence of any information base. No understanding of discipline and its ethical dimension | No evidence of use of learning resources of understanding of self-direction with no evidence of contribution to team work. No evidence academic/intellectual skills and incoherent structure/ expression. No evidence of practical/ professional skills |
| 0% | Awarded for: (i) non-submission; (ii) dangerous practice and; (iii) in situations where the student fails to address the assignment brief and/or related learning outcomes | |