

**Project Marking Scheme (40%)**

|  |  |      |
|--|--|------|
| <b>Report (15%)</b>                      | <ul style="list-style-type: none"> <li>• Class diagram               <ul style="list-style-type: none"> <li>○ Min. 4 classes</li> <li>○ Associations, multiplicity</li> <li>○ Attributes (Min. 3 attributes per class)</li> <li>○ Methods (Min. 3 methods per class)</li> <li>○ Polymorphism                   <ul style="list-style-type: none"> <li>▪ Overriding</li> <li>▪ Overloading</li> </ul> </li> <li>○ Inheritance and/or Aggregation</li> <li>○ Others OO concepts: abstraction, encapsulation</li> </ul> </li> </ul> | = 10 |
|  | • User interface Design (OO rules)   | = 5  |
|  | • Test Case (Min. 5 test cases)  | = 5  |
| <b>Object Oriented Programming (15%)</b> | <ul style="list-style-type: none"> <li>• Reflect class diagram</li> <li>• Implement min 4 classes and 4 objects</li> </ul>   | =2   |
|  | <ul style="list-style-type: none"> <li>• Encapsulation – separate classes and objects</li> <li>• Implement visibility settings</li> </ul>  | =2   |
|  | <ul style="list-style-type: none"> <li>• Abstraction</li> <li>• Min 3 Header files and implementation files separated</li> </ul>   | =3   |
|  | <ul style="list-style-type: none"> <li>• Inheritance</li> <li>• Implement min 1 super class and 1 sub class</li> </ul>   | =2   |
|  | <ul style="list-style-type: none"> <li>• Polymorphism</li> <li>• Min 3 methods overloading</li> <li>• Min 2 methods overriding</li> </ul>  | =3   |
|  | <ul style="list-style-type: none"> <li>• Functional</li> <li>• Add, modify, delete data</li> </ul>   | =2   |
|  | <ul style="list-style-type: none"> <li>• Main Driver (exe)</li> <li>• From one user interface to another</li> </ul>  | =1   |
| <b>Group Presentation (5%)</b>           | • Understanding of the System Design and Prototype   | = 3  |
|  | • Object-oriented concepts projected on the system   | = 2  |
| <b>Individual Contribution (3%)</b>      | • Contribute in class diagram, test case, user interface design and implementation   | =2   |
|  | • Question and Answer  | =1   |
| <b>Group Work (2%)</b>                   | • Active participation in meeting  | =2   |
|  | • Commitment and Effort to attend meeting  |      |
|  | • Supporting other members   |      |
|  | • Contribution to overall work activity  |      |

### UML Diagrams Rubric (Assignment and Project)

| No. | Criteria  | Excellent   | Good  | Average  | Poor  |
|-----|---|---|---|--|---|
| 1   | <b>Class Diagram – Classes</b>                              | Minimum 4 classes are well identified to the case study and demonstrates understanding of why each class falls into the category of Boundary, Entity and Controller. UML notation for class diagram is correct.                 | Minimum 3 classes are well identified and divided into Boundary, Entity and Controller. UML notation for class diagram is correct.  | Classes are identified but not identified properly to Boundary, Entity and Controller.   | No indication of the various classes in the class diagram.  |
|     | <b>Class Diagram – Relationship, Attributes and Methods</b> | All classes are mentioned in the diagram with their methods (return type and parameters), attributes (type and access level) and relationships (Labelled with multiplicity) are well presented. Correct UML notations are used. | All classes are mentioned in the diagram with their methods, attributes and relationships but the layout and presentation is a bit confusing.   | Some classes are in the diagram and a lot of missing methods, attributes and relationships missing.                                  | Class diagram shows no relationship between the various classes.  |
| 2   | <b>User Interface (UI) Design</b>                           | All of the user interfaces reflects the class diagram by applying OO Design Axioms/Corollaries/ Approach/Rules.   | All of the user interfaces reflects the class diagram with some not applying OO Design Axioms/Corollaries/ Approach/Rules.  | Some of the user interfaces reflects the class diagram and some of the UI do not apply OO Design Axioms/Corollaries/ Approach/Rules. | User interfaces are not reflective of the class diagram.  |
| 3   | <b>Test Cases</b>   | Test cases with clearly stated Action, Input Values, Expected Output and Actual Results according to the template provided for at least 2 classes or 3 main methods   | Test cases with minimal missing steps or values for Action, Input Values, Expected Output and Actual Results according to the template provided for less than 2 classes or 3 main methods | Test cases missing either Action, Input Values, Expected Output or Actual Results, still according to the template provided.         | Test cases missing either Action, Input Values, Expected Output or Actual Results and missing values. Do not follow the template. |
| 4   | <b>Report Mechanics</b>                                     | Report has an excellent organization and presents material  | The report can be followed easily and all material is present. Spelling   | There are significant issues with coherence and ordering.  | The report has no apparent organization or logical order.   |

|          |  |   |   |   |   |
|----------|--|---|---|---|---|
|          |  | in a very logical sequence. No spelling or grammar errors, excellent formatting, highly readable. Functionality – working codes with usable user interface, get and set methods   | and grammar errors rare, excellent formatting.  | Significant material is missing. Few pages without mechanics' errors making reading a chore. Readability is a significant problem.  | Mechanics errors make it a struggle to decipher meaning.  |
| <b>5</b> | <b>Presentation with Question and Answer</b> | Answer all questions correctly and confidently to show understanding of the object oriented system design and prototype. Clearly able to pinpoint the contributions of self and members in class diagram, test case, user interface design and implementation.                                    | Answer maximum three questions correctly and clearly to show understanding of the object oriented system design and prototype. Clearly able to pinpoint the contributions of self and members' in class diagram, test case, user interface design and implementation. | Answer minimum two questions correctly to show understanding of the object oriented system design and prototype. Able to pinpoint contributions of self and members' in class diagram, test case, user interface design and implementation.     | Unable to answer questions correctly. Lack of understanding of the object oriented system design and prototype. No confident in pinpointing contributions of self or members in the class diagram, test case, user interface design and implementation. |
| <b>6</b> | <b>OO Program</b>                            | Fully OO program is implemented by reflecting the class diagram (classes implemented in the program) applying OO concepts. The OO program must be able to add, delete and modify data. The user interfaces are well organised and functional: able transition from one user interface to another. | Fully OO program is implemented by reflecting the class diagram (classes implemented in the program) with some parts of the program applying OO concepts. The program can perform two basic data tasks: add, delete and modify. User interfaces are well organised.   | Some of the classes are implemented as OO programming while others used structural programming. Able to perform two basic data tasks: add, delete and modify. User interfaces are clear with some user interfaces not being able to transition. | Structural programming is used instead of OO programming. Only perform one data tasks: add, delete and modify. User interfaces are not well organised.  |