CMP1903M Object Oriented Programming A01 2022-2023

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Learning Outcome** | **Criterion** | **Pass** | **2:2** | **2:1** | **1** |
| [LO1] Demonstrate the use of version control tools in a software development project | Demonstrate the use of version control tools and take part in a code review to peer assess your code (50%)  [Report] | Some evidence is presented to show that you took part in code review.  Your reflections of the process shows a limited engagement with the process.  There is a descriptive assessment of the process. | Evidence is presented to show that you took part in code review.  Your reflections of the process are limited.  There is a descriptive reflection of the process. | Thorough evidence is presented to show that you took part in code review.  Your reflection of the process shows that you have taken account of the reviews by merging your changes.  A more dialogic reflection (‘what if’, etc) is used. | Extensive evidence is presented to show that you took part in code review.  Your reflections of the process are well informed – you show that you have taken effective account of the reviews by merging your changes, or you suggest alternative approaches.  You demonstrate clearly that you have learned from the experience though a critical reflection |
| [LO3] Apply object-oriented principles to the implementation of software programs | Develop an object-oriented solution to a problem (50%)  [Code, Report] | An implementation is presented which is partially complete.  The application works, however its functionality is incomplete. For example only some of the basic requirements are implemented (eg: only ‘Card’ is implemented, but not ‘Pack’).  Limited evidence of object-oriented features such as classes, object instantiation, and method calls.  The checklist is completed. | An implementation is presented which is complete – with issues.  All the basic methods are implemented. The shuffle types are partially implemented.  Erroneous input is handled, but all errors may not be addressed.  Some evidence of object-oriented features such as classes, object instantiation, methods/method calls are present.  The checklist is completed. | An implementation is presented which is complete.  All of the basic methods are implemented. Additional methods are also used.  The shuffle types are implemented as specified in the brief. The Testing class is implemented.  Erroneous input is handled either by error or exception handling methods. i.e. the methods can deal with erroneous input (this is tested with the Testing class)  Evidence of object-oriented features such as classes, object instantiation, encapsulation and methods.  The checklist is completed. | An implementation is presented which is complete..  All of the basic classes are implemented. Additional classes are also used.  Errors are handled either by exception handling or guard clauses. All possible errors are handled.  Extensive evidence of OO features such as (but not limited to) classes, object instantiation, encapsulation , methods and data abstraction.  The checklist is completed. |
| **Weighting is 30% of the module** |  |  |  |  |  |