CMP1903M Object Oriented Programming Assessment 1 2021-2022

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%)	Only some evidence is presented to show that you took part in code review; your description of the process shows a limited engagement with the process.	Clear evidence is presented to show that you took part in code review; your description of the process is limited.	Thorough evidence is presented to show that you took part in code review. Your description of the process is well informed – you show that you have taken account of the reviews by merging your changes.	Extensive evidence is presented to show that you took part in code review; your description of the process is well informed – you show that you have taken effective account of the reviews by merging your changes.
[LO3] Apply object-oriented principles to the implementation of software programs	Develop an object-oriented solution to a problem (50%)	A limited implementation is presented. The application works, however its functionality is incomplete. For example only 'Option 1' is implemented. Erroneous input is handled either by error or exception handling methods, but the errors are not handled completely and/or all possible errors are not handled. Some evidence of object-oriented features such as classes, object instantiation, encapsulation and methods/method calls are present, but they may not be implemented well. The checklist is completed.	An implementation is presented which works. The functionality allows 'Option 1' to be fully implemented. 'Option 2' is attempted – there may be issues with the loading and/or parsing of the file though. Erroneous input is handled either by error or exception handling methods. All errors may not be addressed. Clear evidence of object-oriented features such as classes, object instantiation, encapsulation and methods/method calls are present. The checklist is completed.	An implementation is presented which works. The functionality allows 'Option 1' and 'Option 2' to be fully implemented. Erroneous input is handled either by error or exception handling methods. i.e. the game does not crash with erroneous input. Thorough evidence of object-oriented features such as classes, object instantiation, encapsulation and methods are present. The checklist is completed.	An implementation is presented which works. The functionality allows 'Option 1' and 'Option 2' to be fully implemented and tested with the pre-set example file. Erroneous input is handled either by error or exception handling. 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					