**CMP1902M Object Oriented Programming 2022/23**

**Assignment 2: Report**

[*Expand the sections as necessary*]

Name:

Student ID:

**Application:**

1. **Reflection on the OO features within your code.** (~400 words)
2. **Reflection on your handling of error conditions in your code.** (~200 words)
3. **Reflection on your testing activities: What did you test, and how did you do it?** (~400 words)

*Eg: I tested the application against ….*

1. **Include evidence of the tests *(screenshots are OK)***

**Reflection and Feedback**

1. **What was the most important thing you learned from this assessment?** *(< 200 words)*

*Eg: I learned that If you don't think every day is a good day - try missing a few. You'll see.*

1. **What was the most challenging aspect of this assessment and how did you approach it?** *(<200 words)*

*Eg: I started painting as a hobby when I was little. I didn't know I had any talent. I believe talent is just a pursued interest.*

1. **What would you particularly like to receive feedback on in this assessment?**

**Assignment 2 Checklist**

All of the elements in a section must be checked for it to be considered for that grade (this isn’t guaranteed though). All previous elements must also be complete for a grade to be considered.

Pass standard:

|  |  |
| --- | --- |
| The code compiles and runs. |  |
| Card, Pack, classes are created. |  |
| Class definitions and object instantiation evident. |  |
| Method calls to methods in the same class as ‘Main’ |  |
|  |  |
|  |  |
|  |  |

2:2 standard:

|  |  |
| --- | --- |
| The rules of the tutorial as specified in the brief are implemented. |  |
| Application repeats or quits the game gracefully according to user choice. |  |
| Method calls from ‘Main’ to methods in other classes |  |
| Error handling is evident, some errors are captured, such as (but not limited to), card sums are problematic (don’t add up, are not random etc) |  |
| Class definitions show **encapsulation.** |  |
|  |  |
|  |  |
|  |  |

2:1 standard:

|  |  |
| --- | --- |
| Exception handling is used |  |
| Inheritance showing a class hierarchy |  |
| public/private access control in classes |  |
|  |  |
|  |  |
|  |  |
|  |  |

First standard:

|  |  |
| --- | --- |
| Interfaces are used |  |
| Static and/or Dynamic polymorphism (eg. method overriding) |  |
| Use of virtual/abstract methods |  |
| protected access control is used in class hierarchy |  |
|  |  |
|  |  |
|  |  |
|  |  |