**CMP1902M Object Oriented Programming 2022/23**

**Assignment 1: Report**

[*Expand the sections as necessary*]

Name:

Student ID:

Git/Version Control Repo:

**Code Review**

1. **Who did you provide reviews for?** *(Name, Student ID)*

*James Eyre 26464488*

*Hamza Hamza 26482655*

1. **Who provided reviews for you?** *(Name, Student ID)*

*James Eyre 26464488*

*Hamza Hamza 26482655*

1. **Reflection on code review: What did you consider changing / changed after receiving the reviews?** *(~400 words)*

*Eg: I changed to a simple lift and shift job. Throughput criticality, pre-think our game-plan.*

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

A screenshot of a computer

Description automatically generatedText

Description automatically generated Text

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence

**Reflection and Feedback**

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

The most important things I learned from this assessment were encapsulation and object instantiation. For instantiation it was somewhat difficult to wrap my head around creating objects inside of objects (Cards inside the pack) but once I figured it out it improved my overall understand of object oriented programming. I learned a lot about coding secure classes with getters and setters, and also controlling lists as ReadOnly to avoid people adding cards to it outside of the pack constructor class. Setting a list as private does not stop this feature, so whenever I wanted to manipulate the list of cards in a pack object I passed it as a temporary pack before passing it back. Learning constructors when it comes to classes was also a benefit, and also utilising the override of the ToString method for the class was a valuable lesson.

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

The most challenging aspect of the assessment for me was flow control of the program. I tackled this by breaking down what I wanted each class and method to do before creating it, followed by making sure each method returned the correct information for the name of the method (E.G printing outside of the method with the data returned). Another challenging aspect of flow control was allowing users to decide what they wanted to do, creating menu functionality made this easier but it was a new challenge for me, especially with handling things like the pack running out of cards.

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

I’d like to receive feedback more on my programming practices than anything else. I received a ~70 on my minesweeper assessment even though I had the functionality there, so this time I focused heavily on correct programming practices such as encapsulation and classes (didn’t include those that assessment). Any critique on how to make the code more readable, efficient, or where it could flow better or utilise another class / method would be greatly appreciated as I wasn’t entirely sure where I went wrong with my last one

**Assignment 1: 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:

|  |  |
| --- | --- |
| Received two code reviews. | X |
| Made two code reviews. | X |
| Addressed **at least two** of the questions in each of your reviews. (*1. How is the code documentation, 2. How does the code handle errors, 3. How can the code be improved*) | X |
| The code compiles and runs | X |
| The classes provided by the base code are implemented | X |
| The methods provided by the base code are implemented | X |
| Object instantiation and method calls used in the code. | X |
|  |  |

2:2 standard:

|  |  |
| --- | --- |
| Addressed the **three** questions in each of your reviews. | X |
| There is a short description of the code review process. | X |
| The code compiles and runs | X |
| Some evidence of error handling | X |
|  |  |
|  |  |
|  |  |
|  |  |

2:1 standard:

|  |  |
| --- | --- |
| The code compiles and runs | X |
| No errors evident | X |
| A Test class is used | X |
| There is a description of the code review process | X |
| **Additional** methods to those provided by the base code are implemented – point out with comments | X |
|  |  |
|  |  |
|  |  |

First standard:

|  |  |
| --- | --- |
| Error handling, either by exception handling or other methods is complete and no errors are produced. | X |
| Encapsulation an/or Abstraction are used in the code and identified. | X |
| **Additional** classes to those provided by the base code are implemented – point out with comments |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |