**CMP1902M Object Oriented Programming 2022/23**

**Assignment 2: Report**

[*Expand the sections as necessary*]

Name: Thomas Cole Crawley

Student ID: 26430829

**Application:**

1. **Reflection on the OO features within your code.** (~400 words)

Before I started using C#, I did not know what object oriented principles were, I did not know what they did nor how they helped in making programming easier. When I eventually had to learn what they were, it seemed really daunting to me and I just wanted to stick to the basic way of programming which I was used to in python. However, I was forced into using them whether I liked it or not to complete the assessments for this module and I must say, I do not think I can look at programming the same way ever again.

I truly appreciate the modularity of everything and how it really saves time and effort on not repeating code. It really enforces the DRY mentality which is Don’t Repeat Yourself. Coming into this assessment, I already knew about inheritance and encapsulation which helped tremendously and am now comfortable using those principles but as for polymorphism and interfaces, to say I was overwhelmed is an understatement.

I took my time to try and understand what each of those principles meant and finally went on to use them in my code and it immediately made the process of programming this assessment much easier. It just shows how just putting some effort into understand what the principles mean and how to implement them make your coding experience much better and smoother in the long run. I also used git to track my version history of my code which, as I mentioned in the previous assessment, was an amazing tool. In fact, I actually relied on version control more so on this assessment because I had to try and implement object oriented principles that I didn’t really understand fully to try and make things work and I ended up breaking a lot of my classes and functions. I guess as they say, the best way to learn how to program is to keep breaking your code and figure out how to make it work again.

I must say however, initially I just planned on trying to complete the code without polymorphism and interfaces as I thought they were too out of reach for me but I then slowly found way to incorporate them in little ways and in the end, they added up to make my code more legible and easier to recycle due to the nature of its modularity, which is the true beauty of object oriented programming.

1. **Reflection on your handling of error conditions in your code.** (~200 words)

I think learning from my past assessment for this module, my error handling and foresight has definitely improved since my python days in the first semester. I feel like I am able to think of more ways in which the code could go wrong and so I think about error handling and the many different possibilities of where my code could go wrong. I have found many ways of dealing with errors by looking at other people’s code and as well as researching possible ways on how to handle different unwanted things like input. One of the main structures I use for error handling are if-else statements as they are straight-forward and easy to understand to whoever looks at my code. But I found a new way to handle errors which is called throw New Argument.

This gives the user an error prompt with some dialogue to show what went wrong and I think it’s a really cool way of dealing with the issue however, I much prefer the if-else statements as it takes a reactive approach to dealing with the issue and instead of just telling the user where they went wrong, you can just make your code perform something else when there is an error to save the program form crashing completely which is super handy to have and looks more professional. I also know of the try and catch blocks however, I have not used any in my assessment as the error handling within my code are mainly if-else statements and throw new arguments.

1. **Reflection on your testing activities: What did you test, and how did you do it?** (~400 words)

*Eg: I tested the application against ….*

To test my code to see if it works, I needed to test each of the main elements that run this code. There are two main elements stated in the brief; We had to deal three cards. If you remember in the previous brief, a single card has two values, its value and its suit. It will output something like a Three of Hearts. In this case, however, to turn the card game into a maths tutor, we had to only take one of the two values from the card to produce an output that makes sense. So, for the first term in the equation, we got the value. This represents numbers 1-13. For the second term, we got the suit. This represents the numbers 1-4. So to make a full equation, we needed to have one card that outputs only its value, another card that outputs only its suit, and another card that outputs its value again.

There was one issue with this though. The suits are numbered 1-4, we can’t use that to add, subtract, multiply nor divide so that was a big issue. This stumped me for a while until I realized, I could just make 1 represent the addition sign and make 2 represent the minus sign and so forth. So, I implemented a swap function and it worked.

Now, to test if this deal three cards function properly works and did the mathematics properly, I created a test class and stuck the function in there and called it. This was without asking the user for any kind of input and it does not keep track of the user’s score when testing. I have also made sure to format the output so that it also outputs the answer of the calculation which I was thought looked kind of neat. There was one issue though. I needed C# to calculate these equations properly, which meant it had to implement PEMDAS, which is a mathematical rule which tells you which parts of the equation you need to operate first to get the correct answer. Normally, C# just calculates from left to right but if it did this, it would sometimes produce an incorrect answer which we do not want.

To prevent this from happening, I listed down all the possible combinations of the operations that could go together and made a function that told C# which operations to prioritize first when operating. This saved me so much time and effort in the long run and as well made testing much easier to do. There is another function wherein you had to deal 5 cards instead and PEMDAS was used heavily here as the program has to decide which operation to perform first and it worked as I thought it should. Again, for the deal 5 cards, it was the same method as I used when testing out the deal 3 cards, only PEMDAS was heavily put to work here.

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.*

The most important thing I have learnt from this assessment is to not be afraid of trying new ways to program. I was so stuck in my ways on how I program, thinking that I would never have to change if the way I am doing it is working. However, the code I was producing back then was much more clunky and there was no modularity at all. I made functions do multiple things at once, overcomplicating the program as a whole and making it extremely difficult to problem solve when errors eventually come up, and trust me, they do. Learning new ways on how to program and implementing them in this assessment has been fulfilling to say the least.

Teaching myself to try and break down classes and functions into its most simplest of forms made it so much easier to troubleshoot and recycle code so that there was less repetition than in the last assessment. I usually am afraid of trying new things as I worry that it might be too difficult to comprehend in time for the assessment but these are just coding challenges I must overcome if I am to pursue a career in the field and ultimately, the industry is about change and how you handle it and learn from it to become a better programmer than you were yesterday.

Another thing that may not be a new idea but the usage of Git has completely changed how I program. Since last assessment, it has been a god send to me because if I am forced into trying new things to improve my code and trying new principles, I am bound to break things. Git has given me that safety blanket that allows me to go and explore and try new things without worrying about never recovering my code ever again from a horrible mistake that I made, it’s the peace of mind as a programmer that allows you try different things knowing that there will always be a working version no matter how much you mess up your code, which is extremely comforting.

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.*

As I said earlier in the other sections of this report, the most challenging aspect for me was trying to implement new principles into the way I code and trying to make it work for me. I was so stuck coding the same way as I did years prior that it was so hard to shake off just using encapsulation and inheritance and calling a day. I was forced into using more advance techniques knowing that it would make me a better programmer and have a better understanding of how different code snippets worked. How did I learn how to implement these principles you ask? Well, I learnt how by how everyone learns how to do something, I tested the waters before diving head first. I researched and made my self knowledgeable on what the principles were and made sure that I had a firm understanding of what I was doing and why I was implementing them into my code.

Polymorphism and interfaces were two things that were completely new to me and things that I have never even tried using in any of my coding projects before. It was daunting to say the least. I was worried I was never going to make it in time to submit this assessment because it took me a while to fully grasp what those new principles meant. But eventually I did, and found ways to implement them into my code. I kind of worked backwards in a way, doing what I knew I could do and making sure everything worked and then found little ways on how I could implement polymorphism and interfaces into my code and it is much better with it than without it. I have mentioned this before but the modularity of the code is something that I truly appreciate and will continue to implement in assessments to come.

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

* Error handling
* Better ways I could have implemented the deal cards function
* Better way to implement polymorphism and interfaces
* The principles of OOP

**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 | ✓ |
|  |  |
|  |  |
|  |  |
|  |  |