CMP1902M Object Oriented Programming 2023/24

Assignment 2: Report

[*Expand the sections as necessary*]

Name:

Student ID:

Code repository URL:

Video URL:

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 ….

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

Reflection and Feedback

5. **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.

6. **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.

7. **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.	
Die, Game and Testing classes are created.	
Object instantiation, method calls evident.	
Sevens Out game is created.	
The Testing class is used.	

2:2 standard:

The rules of the Sevens Out game, as specified, 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 erroneous input being made.	
Class definitions show **encapsulation.**	
The Testing class checks the dice sum is correct and that a total of 7 is detected.	
A Statistics class is used	

2:1 standard:

Sevens Out and Three Or More games are implemented.	
Inheritance is implemented, showing a class hierarchy	
`public/private` access control in classes	
Generic collections (such as List<>) are used.	
Exception handling is used	
Testing class uses verification methods in code (such as `debug.assert`()) to check code.	

First standard:

Interfaces and LINQ are used	
Static and/or Dynamic **polymorphism** are evident	
Use of `virtual/abstract` methods	
`protected` access control is used in class hierarchy	
The Testing class implements a way to record testing data (through a log file for example)	