# **CMP2801M Advanced Programming, Fall 2019 - Assessment 1**

**Implemented functionality**

The problem was to create a program that can produce shapes using object orientated programming. This would be approached by setting up the required classes and linking them with virtual functions and objects.

All classes in the brief were implemented (Point, Shape, Movable, Circle, Square, Rectangle) with nothing differing from the brief. No extra methods were added to the classes apart from the point class which contained the setters and getters methods, which were used

- All Circle, Square and Rectangle classes are implemented to derive from both Shape and Movable classes.

-Virtual functions were used for all methods in Shape and Movable

- No error checking was implemented

- No additional features were implemented

**Test Plan and Test Cases**

The program has been tested to make sure all shapes can be made, moved and scaled corresponding to user input. As all outputs are correct this shows that all classes, abstract classes, methods and virtual functions have been implemented correctly. It also shows all inheritance and encapsulation was implemented correctly.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input Values** | **Expected output** | **Passed?** |
| Add rectangle valid values | leftTopx 100  leftTopy 100  height 50  width 200 | Object created with correct values, i.e.  Rectangle [h= 50, w= 200]  (100, 100)  (300, 100)  (300, 150)  (100, 300)  Area = 10000 Perimeter = 500 | YES |
| Add square valid values | leftTopx 200  leftTopy 250  edge 30 | Object created with correct values  Square edge = 30  (200, 250)  (230, 250)  (230, 280)  (200, 280)  Area = 900 Perimeter = 120 | YES |
| Add square invalid edge value | leftTopx 200  leftTopy 250  edge -30 | Object with wrong values  Square edge = -30  (200, 250)  (170, 250)  (170, 220)  (200, 220)  Area = 900 Perimeter = -120 | YES |
| Add Circle with valid values | leftTopx 100  leftTopy 150  radius 20 | Object with correct values  Circle radius = 20.000000  (100, 150)  (2040, 3040)  area = 1256  perimeter = 125 | YES |
| Move a circle | move 1 200 200 | Circle that has been moved to top left point 200 200  Circle radius = 20.000000  (200, 200)  (4040, 4040)  area = 1256  perimeter = 125 | YES |
| Scale a circle | Scale 1 2 2 | Circle that has been scaled by 2  Circle radius = 40.000000  (100, 150)  (4080, 6080)  area = 5024  perimeter = 251 | YES |
| Move a square | Move 1 300 500 | Square that as been moved to 300 500  Square edge = 30  (300, 500)  (330, 500)  (330, 530)  (300, 530)  Area = 900 Perimeter = 120 | YES |
| Scale a square | Scale 1 3 3 | Square scaled by 3  Square edge = 90  (200, 250)  (290, 250)  (290, 340)  (200, 340)  Area = 8100 Perimeter = 360 | YES |

Program analysis

The program could be a lot more user friendly and efficient with the use of exception handling. This would stop the program from crashing if a wrong input is inputted. The scaling of circles could have been done so scaling by 2 doubled the size of the circle instead of by 4 times.