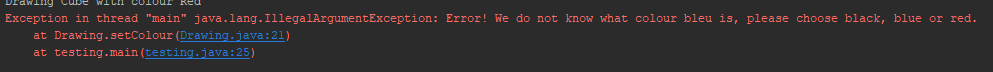
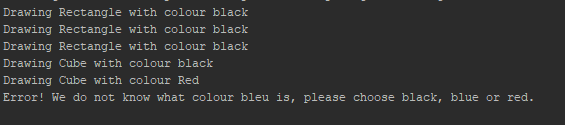
# CT2106 OOP – Assignment 5

# By Luke Gibbons – 25/11/2018

Project Output When Exception Occurred:



Project Output After Exception Try and Catch Was Implemented:



How My Project Works?

To achieve a composite design pattern, the assignment layout is very like that of the Files/Folders in the lectures. So the Shape class acts as the abstract class. Drawings classes (like folders) extend this class as well as the shapes Cone, Cube and Rectangle (which act like the files).

There was no need to code the body of the abstract class or the Cube, Cone and Rectangle classes in order for the output to be achieved, so all the working are from within the Drawing class.

The add and remove methods work through the use of an Array created in the Drawing class. This array can hold both drawings and cone, cube and rectangle shapes due to them all extending the abstract Shape class.

The print method is also very simple as it involves looping through the Shape object array and outputting a string.

Finally, the setColour method works by taking in a string argument and checking if it is black, blue or red through the use of colour.equalsIgnoreCase(“”); The method also throws an exception using ‘throw new IllegalArgumentException’ if the colour does not match with the ones available. I implemented a catch if an exception is thrown in the testing class. This works by trying the setColour method and if an exception is thrown the program will catch it and output the error method.