Project 1

By Matthew Jones

Contents

[Introduction 3](#_Toc498085458)

[Testing 3](#_Toc498085459)

[Testing Table 6](#_Toc498085460)

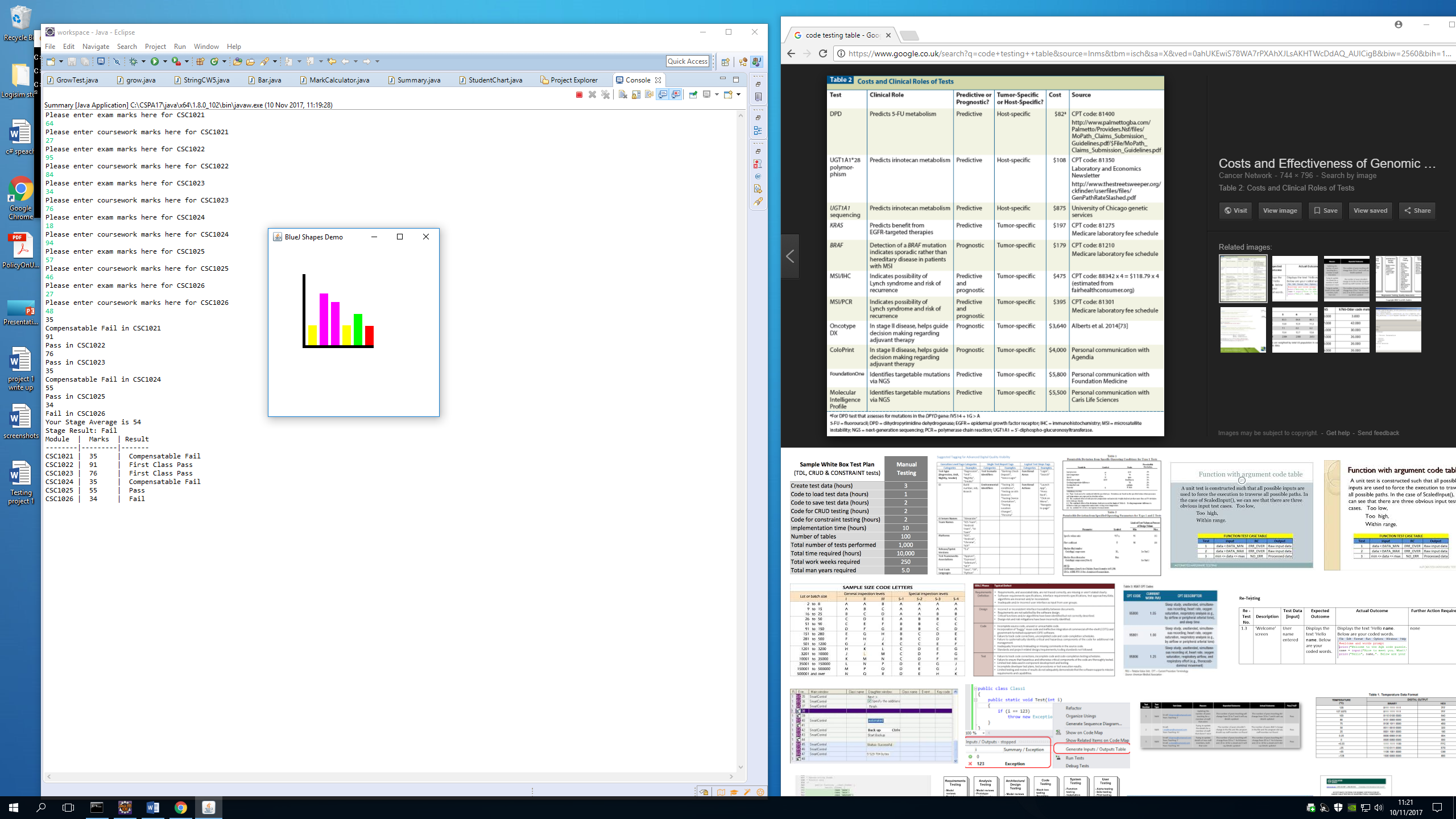
[Evaluation 6](#_Toc498085461)

# Introduction

In this project I was asked to make a program which produces module marks and a stage result, after being given a student’s exam and coursework marks. After this, I had to create a corresponding bar chart and table which would display these results. Lastly, I had to create a summary class which would run all of the code neatly together.

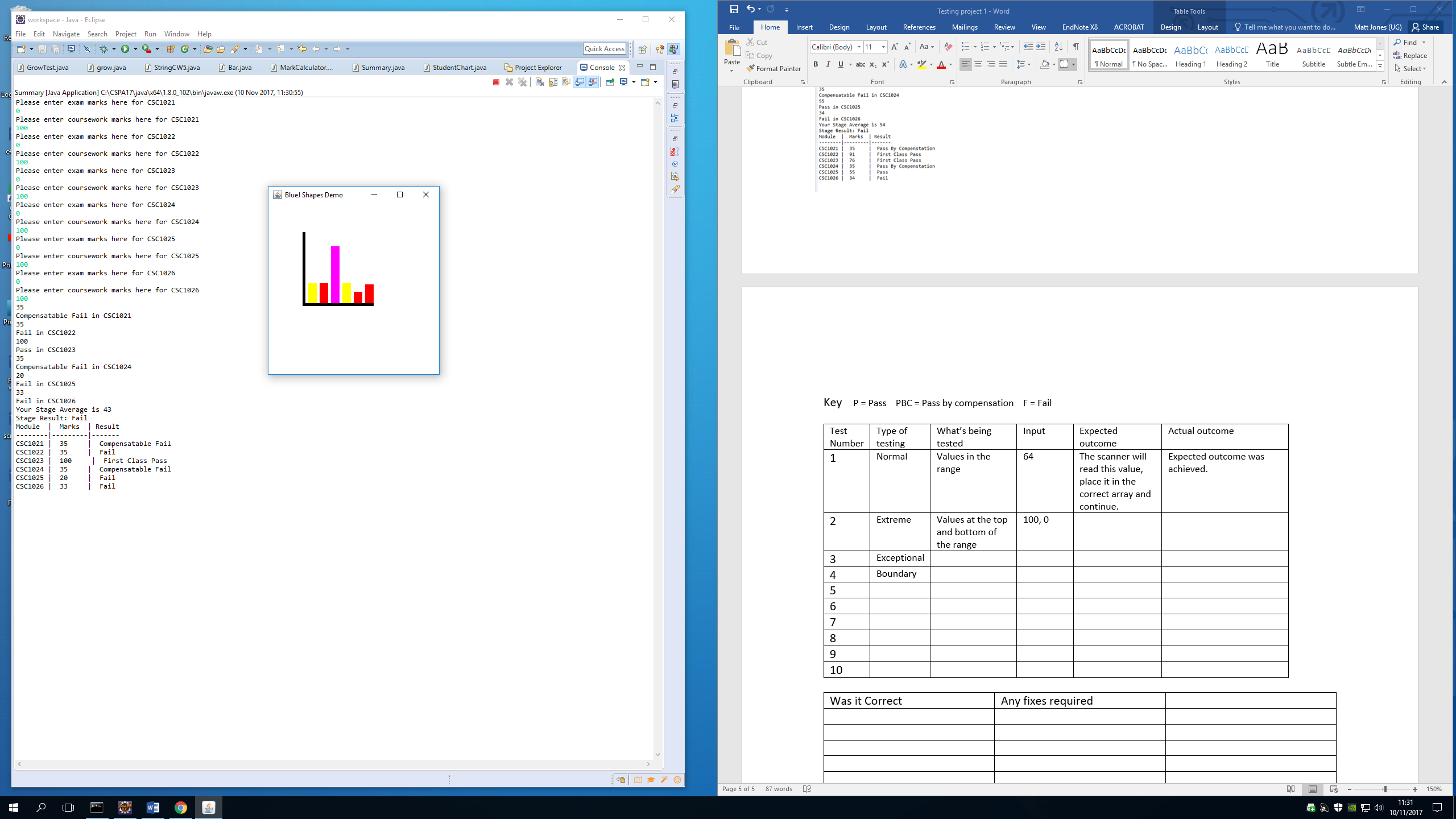
# Testing

Test 1

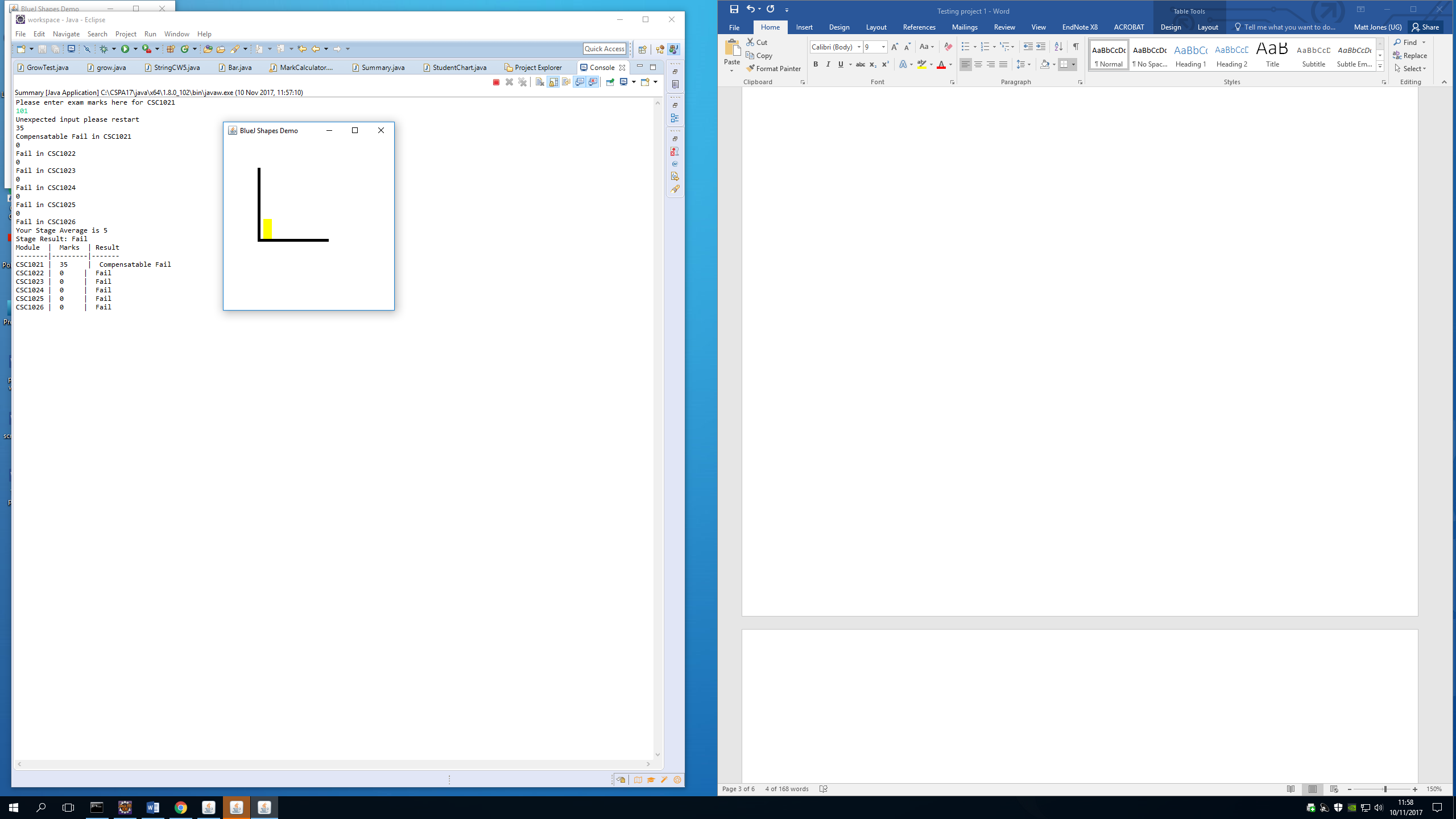


In this test, I was input expected values from a student’s marks using the code to the right. The test allows all possible values that a student would achieve in the modules, and runs through without a problem. The first module is capped at 35, because one of the student’s marks was lower than 35 and the student still achieves an average higher then 35. The last module is recorded as a fail, because the average was not higher then 35; as a result of this the student has achieved a fail as their Stage Result. The colour coded bar chart clearly shows their module marks. The table also displays their module marks and the result for that module.

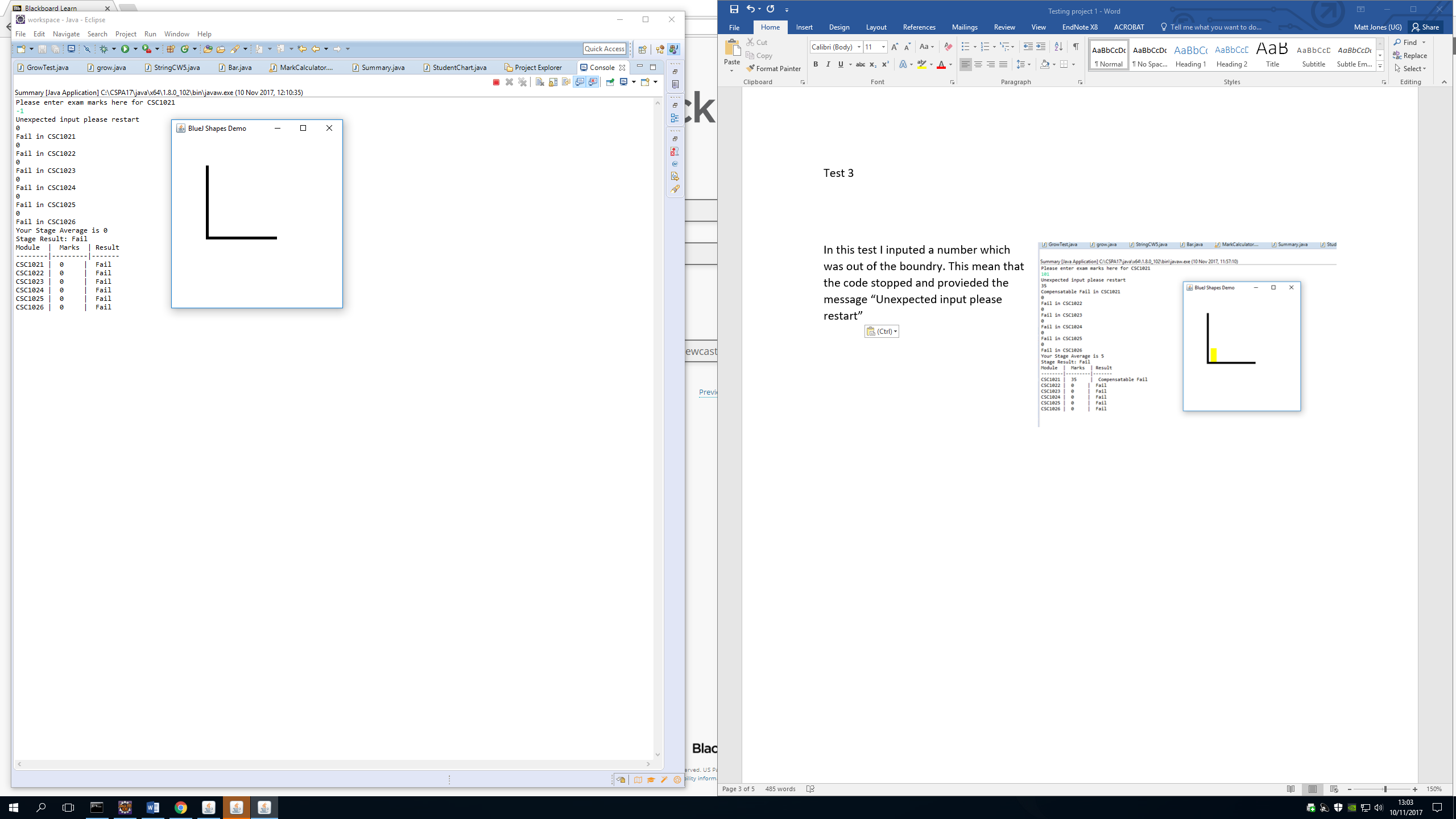
Test 2

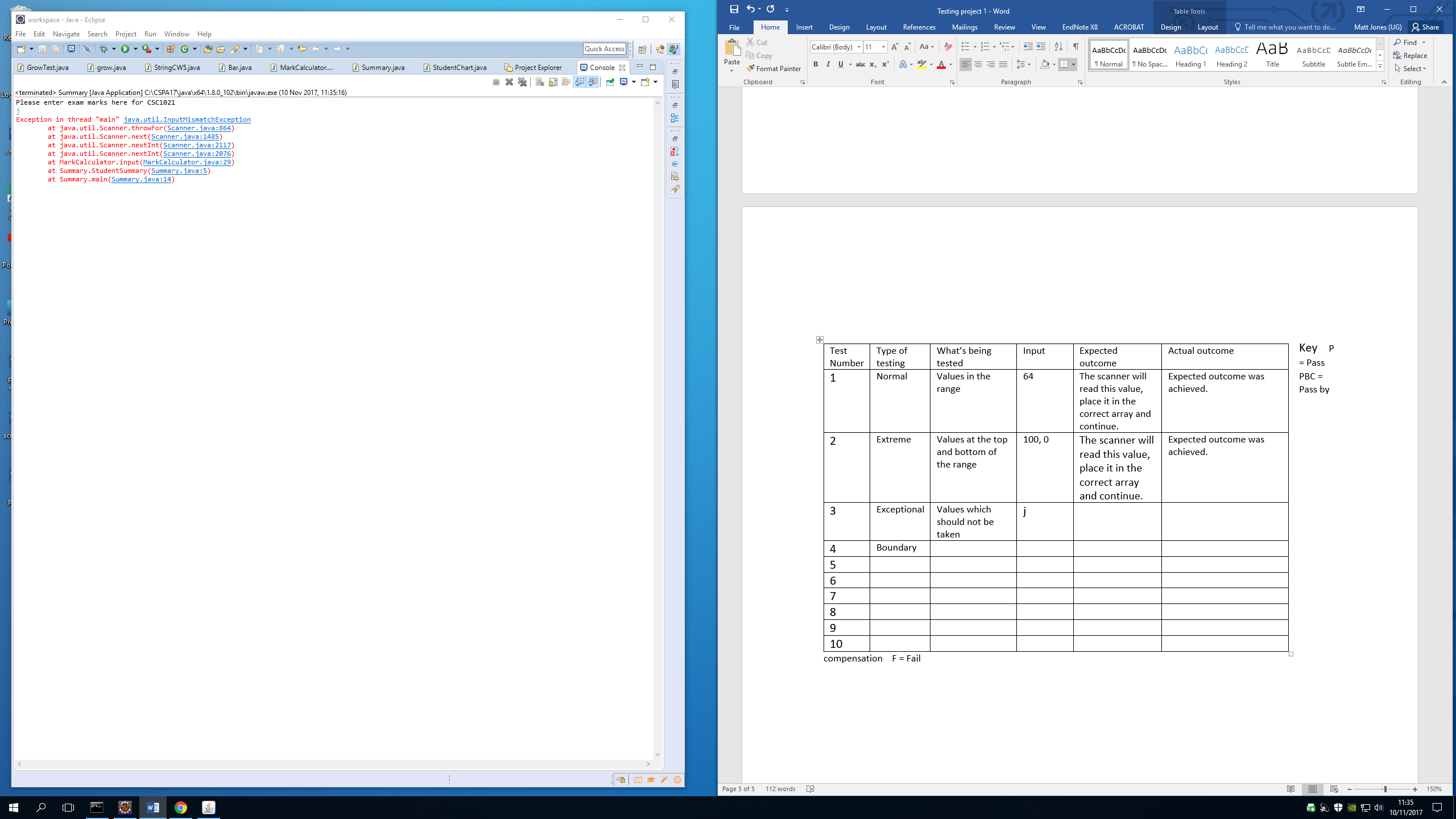
In this test, I used the boundary integers from 0 to 100. Most of the modules are capped at 35, because the average is higher than 35 and results in a compensatable fail. The CSC1023 module only requires a coursework mark so the exam mark input is disregarded. Furthermore, CSC1025 has a coursework weighting of 0.20 having a value of only 20% and average of only 20. This means that the value is not capped.

As there was a fail, the stage result is a fail. This can be seen on the chart and table to the right.

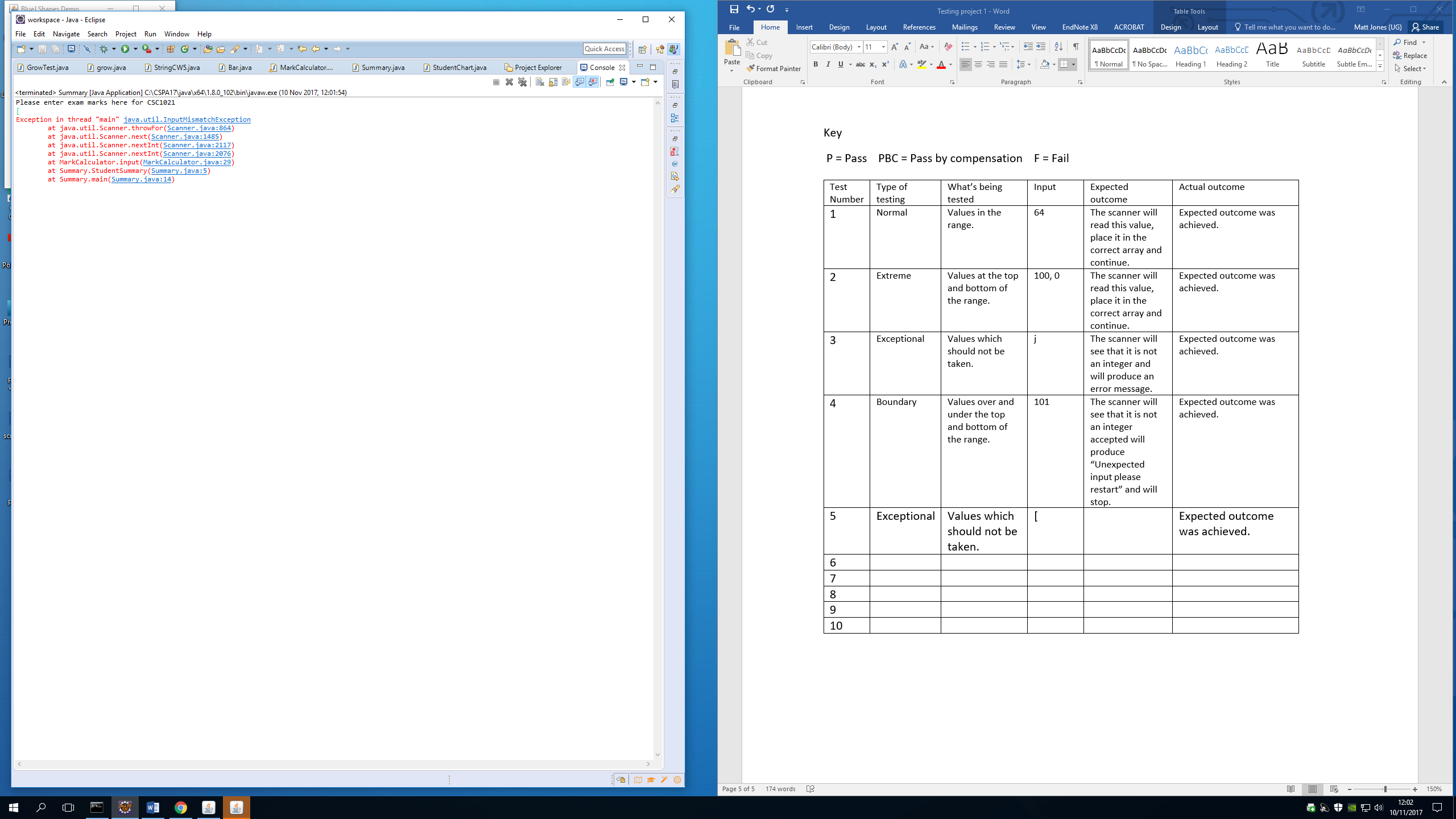
Test 3

In this test, input integers are outside of the boundry. This means that the code was stopped and provided the message “Unexpected input please restart”. The user would then restart the code and enter vaild integers.

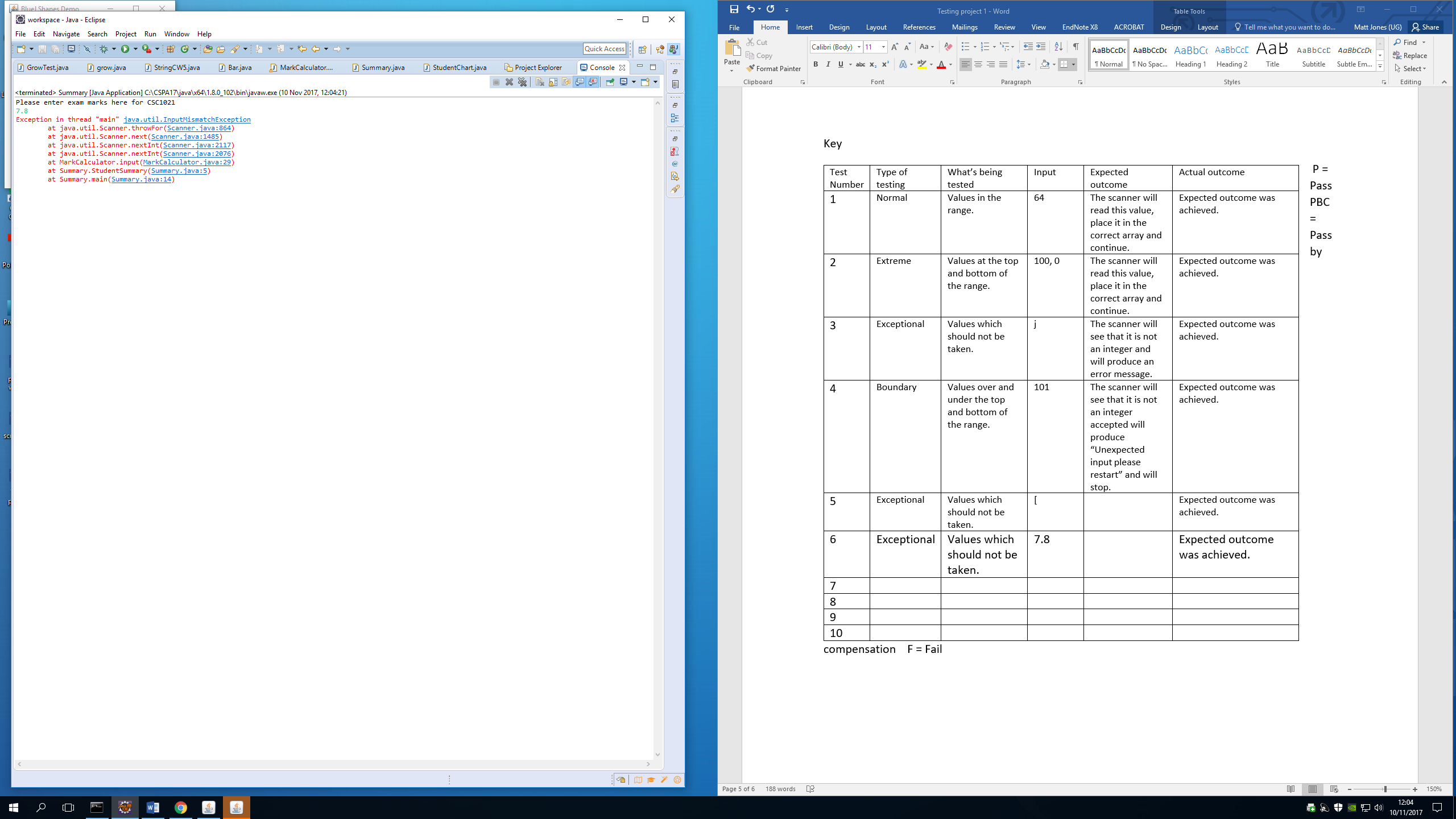


Test 4

With this test, values were input which were not integers. The code does not accept this and an error code is produced. The user would then restart the code and proceed to type in valid values instead.

Test 5

With this test, values were input which were not integers. The code does not accept this and an error code is produced. The user would then restart the code and proceed to type in valid values instead.

Test 6

In the following code, the values entered were doubles. The code does not accept these values and an error code is produced. The user would then restart the code and proceed to type in valid values instead.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Type of testing | What’s being tested | Input | Expected outcome | Actual outcome |
| 1 | Normal | Values in the range. | “64” | The scanner will read this value, place it in the correct array and continue. | Expected outcome was achieved. |
| 2 | Extreme | Values at the top and bottom of the range. | “0”  “100” | The scanner will read this value, place it in the correct array and continue. | Expected outcome was achieved. |
| 3 | Boundary | Values over and under the top and bottom of the range. | “101” | The scanner will see that it is not an integer which will produce “Unexpected input please restart” and will stop. | Expected outcome was achieved. |
| 4 | Exceptional | Values which should not be taken. | “j” | The scanner will see that it is not an integer and will produce an error message. | Expected outcome was achieved. |
| 5 | Exceptional | Values which should not be taken. | “[“ | The scanner will see that it is not an integer and will produce an error message. | Expected outcome was achieved. |
| 6 | Exceptional | Values which should not be taken. | “7.8” | The scanner will see that it is not an integer and will produce an error message. | Expected outcome was achieved. |

# Testing Table

# Evaluation

An improvement which could have been made if more time was available, would have been complete validation for values which were not integers by producing a message and stopping the code, instead of the error message shown. This would increase efficacy and create a higher professional look to the code. Another improvement would be to change the table to line up despite how many characters were produced. In conclusion, I am pleased with the outcomes of this project and I believe that the specification requirements were appropriately met.