|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Purpose | Expected Outcome | Actual Outcome | Changes Made |
| 1 | Test the ability to detect input values <0 | If an input value is below 0, a statement will be printed informing the user this is not valid, along with a message to rerun the program and termination of the program. | The statement is printed, a message to rerun the program is printed and the process is terminated. | None |
| 2 | Test the ability to detect input values >100 | If an input value is above 100, a statement will be printed informing the user this is not valid, along with a message to rerun the program and termination of the program. | The statement is printed, a message to rerun the program is printed and the process is terminated. | None |
| 3 | Values entered produce the correct output mark when ran through the equation. | I will enter “65” and “40” for the 1st module, this should output the value “53” (rounded to nearest whole). The equation for this is ((((40 \* 50) + (65 \* (100 - 50))) + 50) / 100) | 53 is output. | None |
| 4 | Values entered produce the correct output mark when ran through the equation. | I will enter “65” and “40” for the 2nd module, this should output the value “55” (rounded to nearest whole). The equation for this is ((((40 \* 40) + (65 \* (100 - 40))) + 50) / 100) | 55 is output. | None |
| 5 | Values entered produce the correct output mark when ran through the equation. | I will enter “65” and “40” for the 4th module, this should output the value “53” (rounded to nearest whole). The equation for this is ((((40 \* 50) + (65 \* (100 - 50))) + 50) / 100) | 53 is output. | None |
| 6 | Values entered produce the correct output mark when ran through the equation. | I will enter “65” and “40” for the 5th module, this should output the value “60” (rounded to nearest whole). The equation for this is ((((40 \* 20) + (65 \* (100 - 20))) + 50) / 100) | 45 is output.  If 2 values are entered for CSC1023 when only one should be entered, the second value carries over to the next input as a scanner token, this means when the user enters two values for CSC1024, only the first is used and enters the array as the 1st element instead of 0th | Added next() after asking for the CSC1023 mark along with asking the user to enter any character to confirm, this acts as a flush for the scanner |
| 7 | Values entered produce the correct output mark when ran through the equation. | I will enter “65” and “40” for the 5th module, this should output the value “60” (rounded to nearest whole). The equation for this is ((((40 \* 20) + (65 \* (100 - 20))) + 50) / 100) | Thanks to the fix in the previous test, 60 is output. | None |
| 8 | Values entered produce the correct output mark when ran through the equation. | I will enter “65” and “40” for the 6th module, this should output the value “56” (rounded to nearest whole). The equation for this is ((((40 \* 35) + (65 \* (100 - 35))) + 50) / 100) | 56 is output. | None |
| 9 | Resulting marks give a Pass result. | Using the marks {40, 40, 40, 40, 40, 40}, the resulting grade for each module should be Pass. | The results are correct for each module. | None |
| 10 | Resulting marks give a Compensatable Fail (CFail) result. | Using the marks {37, 37, 37, 37, 37, 37}, the resulting grade for all but CSC1022 should be CFail, CSC1022 should be Fail. | The results are correct for each module. | None |
| 11 | Resulting marks give a Fail result. | Using the marks {20, 20, 20, 20, 20, 20}, the resulting grade for each module should be Fail. | The results are correct for each module. |  |
| 12 | Bar graph colour matches the result. | Using the marks {70, 70, 70, 70, 70, 70}, all bars on the bar graph should be magenta to indicate a First class mark. | All bars are magenta. | None |
| 13 | Bar graph colour matches the result. | Using the marks {40, 40, 40, 40, 40, 40}, all bars on the bar graph should be green to indicate a pass. | All bars are green. | None |
|  |  |  |  |  |
| 14 | Bar graph colour matches the result. | Using the marks {37, 37, 37, 37, 37, 37}, aside from the second bar which should be red, all other bars should be yellow to indicate a Compensatable Fail | All bars except the second one are yellow, the second one is red. | None |
| 15 | Bar graph colour matches the result. | Using the marks {20, 20, 20, 20, 20, 20}, all bars should be red to indicate a fail. | All bars are red. | None |
| 16 | Stage Result is correct. | Using the values {45, 45, 45, 45, 45, 45} and {"Pass", "Pass", "Pass", "Pass", "Pass", "Pass"}, the Stage Result should be Pass. | Stage result is Pass. | None |
| 17 | Stage Result is correct. | Using the values {45, 45, 37, 37, 45, 45} and {"Pass", "Pass", "CFail", "CFail", "Pass", "Pass"}, the Stage Result should be Pass By Compensation. | Stage Result is Pass By Compensation. | None |
| 18 | Stage Result is correct. | Using the values {45, 45, 37, 37, 37, 45} and {"Pass", "Pass", "CFail", "CFail", "CFail", "Pass"}, the Stage Result should be Fail. | Stage Result is Fail. | None |
| 19 | Stage Result is correct. | Using the values {45, 37, 45, 45, 45, 45} and {"Pass", "Fail", "Pass", "Pass", "Pass", "Pass"}, the Stage Result should be Fail. | Stage Result is Fail. | None |