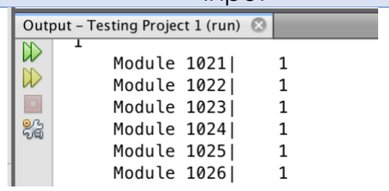
Programming I - 1021

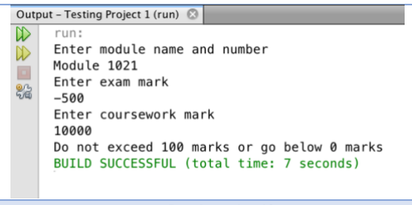
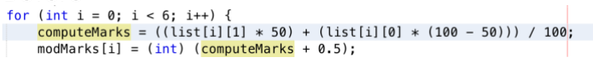
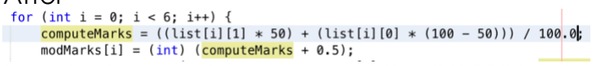
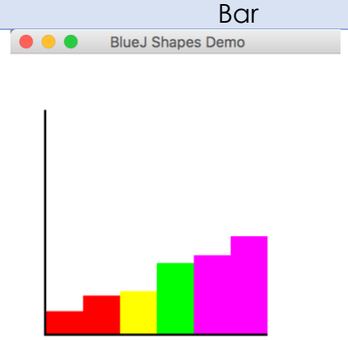
Project 1 – Mark Calculator

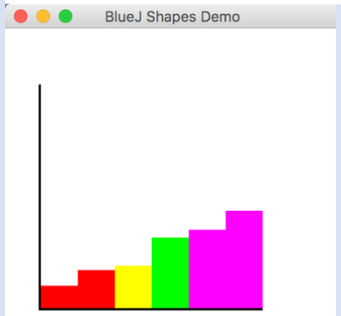
By Ibrahim Ali -

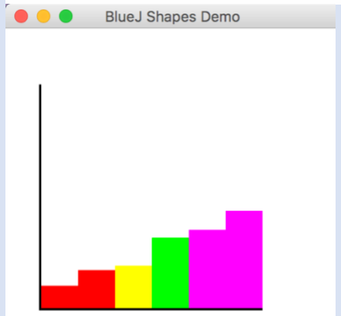
Testing – Project 1

Test	Input	Expected Output	Output	Fixes/Changes	Description
			Input		
Does the program allow the user to input up to six modules?	Module 1021 Module 1022 Module 1023 Module 1024 Module 1025 Module 1026	"Module 1021 | Module 1022 | Module 1023 | Module 1024 | Module 1025 | Module 1026 |"		N/A	The program allows up to six modules, as required per specification.
Does the program allow the user to input the exam marks (testing user input) and produce an output from these marks (testing output).	23 84 0 100 27 89	"Enter exam mark 23 Enter exam mark 84 Enter exam mark 0 Enter exam mark 100 Enter exam mark 27 Enter exam mark 89"	Enter module name and number Mod 1 Enter exam mark 23 Enter coursework mark 56 Enter module name and number Mod 2 Enter exam mark 84 Enter coursework mark 28 Enter module name and number Mod 3 Enter exam mark 0 Enter coursework mark 87 Enter module name and number Mod 4 Enter exam mark 100 Enter coursework mark 76 Enter module name and number Mod 5 Enter exam mark 37 Enter coursework mark 38 Enter module name and number Mod 6 Enter exam mark 89 Enter coursework mark 12	N/A	By inputting the exam marks this tests that the user input functionality works, and to evidence the inputs have been accepted the output should show what is expected in 'Expected Output'.

Does the program allow the user to input the coursework marks (testing user input) and produce an output from these marks (testing output).	56 28 87 76 38 12	"Enter coursework mark 56 Enter coursework mark 28 Enter coursework mark 87 Enter coursework mark 76 Enter coursework mark 38 Enter coursework mark 12"	Enter module name and number Mod 1 Enter exam mark 23 Enter coursework mark 56 Enter module name and number Mod 2 Enter exam mark 84 Enter coursework mark 28 Enter module name and number Mod 3 Enter exam mark 0 Enter coursework mark 87 Enter module name and number Mod 4 Enter exam mark 100 Enter coursework mark 76 Enter module name and number Mod 5 Enter exam mark 37 Enter coursework mark 38 Enter module name and number Mod 6 Enter exam mark 89 Enter coursework mark 12	N/A	By inputting the coursework marks this tests that the user input functionality works, and to evidence the inputs have been accepted the output should show what is expected in 'Expected Output'.
	Validation				
Does the program have validation to ensure the integers aren't less than zero?	Exam mark: -50 Coursework mark: 87	"Do not exceed 100 marks or go below 0 marks" The program should then terminate itself.	Output - Testing Project 1 (run) run: Enter module name and number Module 1021 Enter exam mark -50 Enter coursework mark 87 Do not exceed 100 marks or go below 0 marks BUILD SUCCESSFUL (total time: 15 seconds)	Despite the validation running after both marks for a module have been entered, I've deemed it something that doesn't need fixing as the validation still works.	
Does the program have validation to ensure the integers aren't greater than 100?	Exam mark: 69 Coursework mark: 101	"Do not exceed 100 marks or go below 0 marks" The program should then terminate itself.	Output - Testing Project 1 (run) run: Enter module name and number Module 1021 Enter exam mark 69 Enter coursework mark 101 Do not exceed 100 marks or go below 0 marks BUILD SUCCESSFUL (total time: 15 seconds)	N/A	

Will the program terminate if multiple integers exceed the validation?	Exam mark: -500 Coursework mark: 10000	"Do not exceed 100 marks or go below 0 marks" The program should then terminate itself.		N/A	
	Calculations				
Is the computed module mark computed correctly?	Module | Exam | Coursework 1021 | 20 | 14 1022 | 34 | 32 1023 | 36 | 39 1024 | 56 | 69 1025 | 70 | 69 1026 | 90 | 83	Module 1021| 17 Module 1022| 33 Module 1023| 38 Module 1024| 63 Module 1025| 70 Module 1026| 87	Module 1021| 17 Module 1022| 33 Module 1023| 37 Module 1024| 62 Module 1025| 69 Module 1026| 86	Before  After  In order to round up I ensured the "/100" had a ".0" at the end of it, so it would produce a double value. Rather than the int it was producing before.	
Is the computed module mark computed correctly after the change?	Module | Exam | Coursework 1021 | 20 | 14 1022 | 34 | 32 1023 | 36 | 39 1024 | 56 | 69 1025 | 70 | 69 1026 | 90 | 83	Module 1021| 17 Module 1022| 33 Module 1023| 38 Module 1024| 63 Module 1025| 70 Module 1026| 87	Module 1021| 17 Module 1022| 33 Module 1023| 38 Module 1024| 63 Module 1025| 70 Module 1026| 87	N/A	
Does the program round up the computed module mark results to the nearest whole number?	Module | Exam | Coursework 1023 | 20 | 14 1024 | 56 | 69 1025 | 70 | 69 1026 | 90 | 83	Module 1023| 38 Module 1024| 63 Module 1025| 70 Module 1026| 87	Module 1023| 38 Module 1024| 63 Module 1025| 70 Module 1026| 87	This was previously fixed when testing the computed module mark.	
	Bar				
Do the bar colours change depending on the module mark? (As per the specification)	Module | Exam | Coursework 1021 | 13 | 27 1022 | 34 | 34 1023 | 36 | 39 1024 | 56 | 69 1025 | 70 | 69 1026 | 90 | 83	1021 – RED 1022 – RED 1023 – YELLOW 1024 – GREEN 1025 – MAGENTA 1026 - MAGENTA		N/A	
	Print Summary				

<table>
<tr>
<td>Does the print summary print the returned marks for a module?
Does the print summary neatly output the returned marks?</td>
<td>
<table>
<tr><th>Module</th><th>Exam</th><th>Coursework</th></tr>
<tr><td>1021</td><td>13</td><td>27</td></tr>
<tr><td>1022</td><td>34</td><td>34</td></tr>
<tr><td>1023</td><td>36</td><td>39</td></tr>
<tr><td>1024</td><td>56</td><td>69</td></tr>
<tr><td>1025</td><td>70</td><td>69</td></tr>
<tr><td>1026</td><td>90</td><td>83</td></tr>
</table>
</td>
<td>
<pre>
Module 1021| 20
Module 1022| 34
Module 1023| 38
Module 1024| 63
Module 1025| 70
Module 1026| 87
</pre>
</td>
<td>
<pre>
Module 1021| 20
Module 1022| 34
Module 1023| 38
Module 1024| 63
Module 1025| 70
Module 1026| 87
</pre>
</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Does the print summary print out returned marks that correspond to the chart?</td>
<td>
<table>
<tr><th>Module</th><th>Exam</th><th>Coursework</th></tr>
<tr><td>1021</td><td>13</td><td>27</td></tr>
<tr><td>1022</td><td>34</td><td>34</td></tr>
<tr><td>1023</td><td>36</td><td>39</td></tr>
<tr><td>1024</td><td>56</td><td>69</td></tr>
<tr><td>1025</td><td>70</td><td>69</td></tr>
<tr><td>1026</td><td>90</td><td>83</td></tr>
</table>
</td>
<td>
<pre>
Module 1021| 20
Module 1022| 34
Module 1023| 38
Module 1024| 63
Module 1025| 70
Module 1026| 87
</pre>


</td>
<td>
<pre>
Module 1021| 20
Module 1022| 34
Module 1023| 38
Module 1024| 63
Module 1025| 70
Module 1026| 87
</pre>


</td>
<td>N/A</td>
<td></td>
</tr>
</table>

How you know it works correctly?

I know that my program works correctly as the above testing has rectified any possible errors and ensures it meets the specification.

Each part of the test was to ensure that I have completed stages required by the specification. Each test ensures that the program is functional but is also compared to the specification to ensure the expected outcomes are the same. This is to prevent any deviation, allowing for the possibility that the program is not as the specification.

The testing has evidenced that my program works, screenshots have also been provided.