# Test Document

Table of Contents

User Menu Tests ---------------------------------------------------------------------------------------------------------------------------------

* Test 1: ‘Would you like to start?’ with valid input
* Test 2: ‘Would you like to start?’ with valid extreme input
* Test 3: ‘Would you like to start?’ with invalid input
* Test 4: ‘Select a menu option:’ with valid inputs
* Test 4.a: Menu option fullListOfStudents()
* Test 4.b: Menu option enterNewRecord()
* Test 4.c: Menu option editAMark()
* Test 4.d: Menu option deleteARecord()
* Test 4.e: Menu option generateStatisticsReport()
* Test 5: ‘Select a menu option:’ with valid extreme inputs.
  + Test 5.a: Menu option fullListOfStudents()
  + Test 5.b: Menu option enterNewRecord()
  + Test 5.c: Menu option editAMark()
  + Test 5.d: Menu option deleteARecord()
  + Test 5.e: Menu option generateStatisticsReport()
* Test 6: ‘Select a menu option:’ with invalid inputs.
* Test 7: ‘Would you like to continue:’ with valid inputs – ‘Yes’
* Test 8: ‘Would you like to continue:’ with valid inputs – ‘No’
* Test 9: ‘Would you like to continue:’ with invalid inputs

Full List of Students Tests ----------------------------------------------------------------------------------------------------------------------

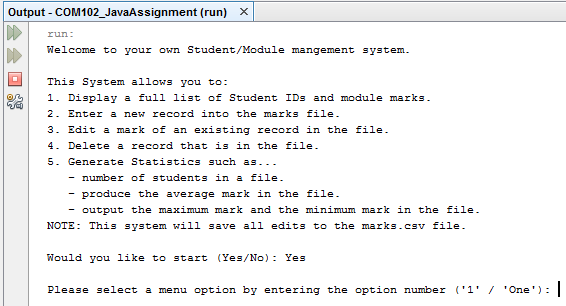
* Test 10: Displaying a full list of correct records

Enter a new record Tests -----------------------------------------------------------------------------------------------------------------------

* Test 11: ‘Enter the student number:’ with valid inputs
* Test 12: ‘Enter the student number:’ with invalid inputs
  + Test 12.a: ‘Enter the student number:’ with invalid inputs
  + Test 12.b: ‘Enter the student number:’ with invalid inputs
* Test 13: ‘Incorrect student number, Please try again:’ with valid inputs
* Test 14: ‘Incorrect student number, Please try again:’ with invalid inputs
* Test 15: ‘Enter the student mark:’ with valid inputs
* Test 16: ‘Enter the student mark:’ with valid extreme inputs
* Test 17: ‘Enter the student mark:’ with invalid inputs
* Test 18: ‘Invalid data, would you like to try again:’ with valid inputs
* Test 19: ‘Invalid data, would you like to try again:’ with valid extreme inputs
* Test 20: ‘Invalid data, would you like to try again:’ with invalid inputs
* **Edit a Mark Tests** ----------------------------------------------------------------------------------------------------------------------
* Test 21: ‘Select the record you would like to edit:’ with valid results
* Test 22: ‘Select the record you would like to edit:’ with invalid results
* Test 23: ‘Is that the correct value?’ with valid results
* Test 24: ‘Insert a new mark:’ with valid results
* Test 25: ‘Insert a new mark:’ with valid extreme results
* Test 26: ‘Insert a new mark:’ with invalid results
* **Delete a Record Tests** ----------------------------------------------------------------------------------------------------------------
* Test 27: ‘Select the record you would like to Delete:’ with valid results
* Test 28: ‘Select the record you would like to delete:’ with invalid results
* Test 29: ‘Is that the correct value?’ with valid results
* Test 30: ‘Is that the correct value?’ with valid extreme results
* **Generate a Statistics Report** -------------------------------------------------------------------------------------------------------
* Test 31: ‘Select a menu option:’ with valid inputs
* Test 31.a: Menu option numOfRecords ()
* Test 31.b: Menu option averageMark ()
* Test 31.c: Menu option minAndMax ()
* Test 32: ‘Select a menu option:’ with valid extreme inputs.
* Test 33: ‘Select a menu option:’ with invalid inputs.

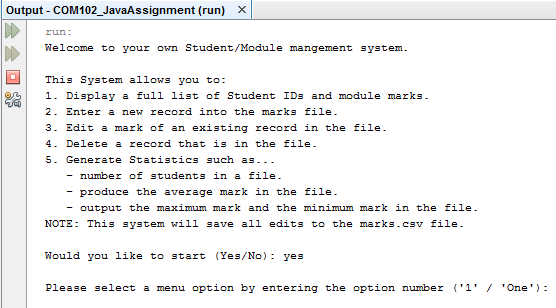
## Test 1: ‘Would you like to start?’ with valid input

This demonstration is to test whether a user can proceed to use the system after being prompted to enter the expected input



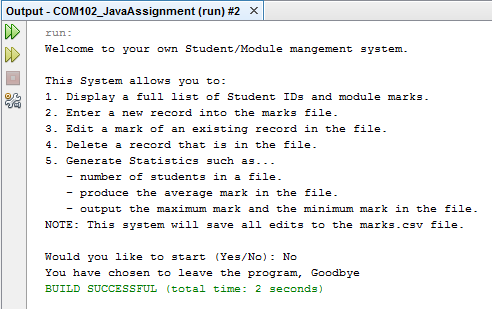
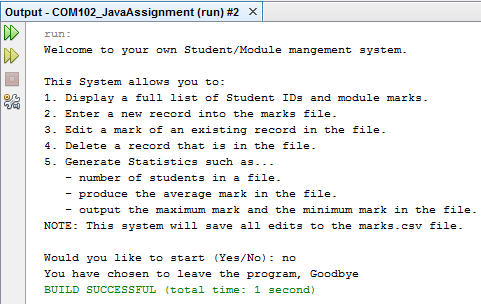
As seen above, if the user enters the expected “Yes”, he/she can continue using the system.

If the user inputs “yes” with a lowercase ‘y’, they are also able to continue. See below.



Obviously, with the user being prompted to enter “Yes” or “No”, another form of valid input would be to test if “No” or “no” will stop the system from proceeding.

We see below that both variations of “No” will exit the system.

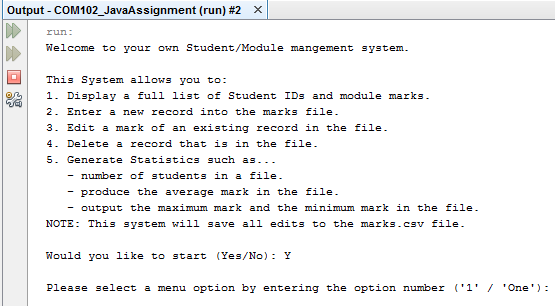


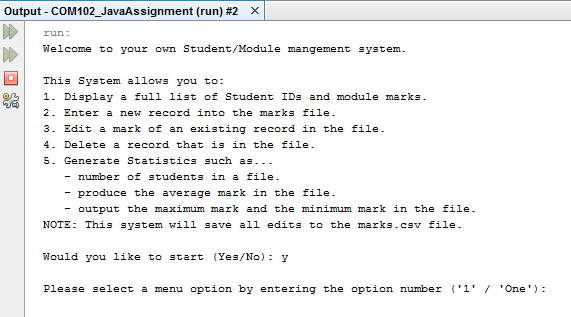
## Test 2: ‘Would you like to start?’ with valid extreme input

This demonstration is to test whether a user can proceed to use the system after being prompted to enter an unexpected but understandably valid input.

Whether the user enters an uppercase or lowercase ‘Y’, he/she will be able to proceed with the system.

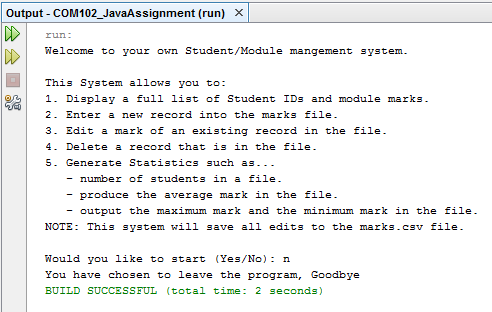
Since they are unexpected, ‘Y’ and ‘y’ both fall under the umbrella of valid extreme input. This is because when asked to start, the user is specifically told to enter “Yes” or “No”.

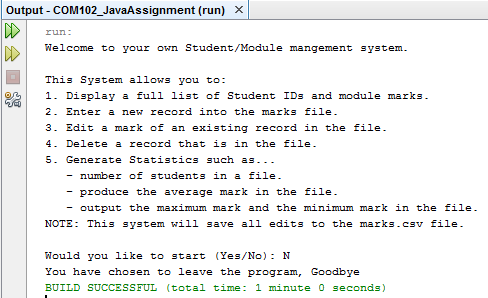




The same applies for “n” and “N” when the user wants to exit the system:

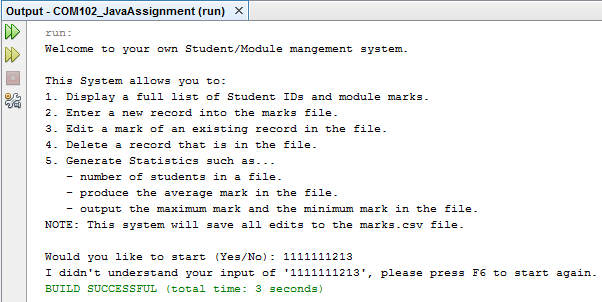
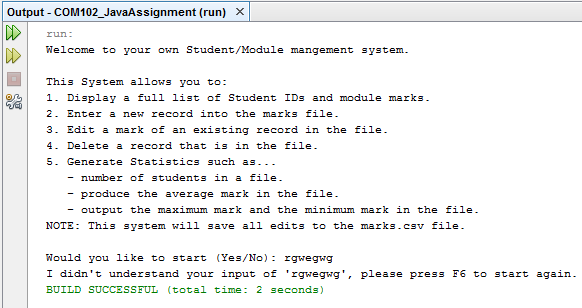
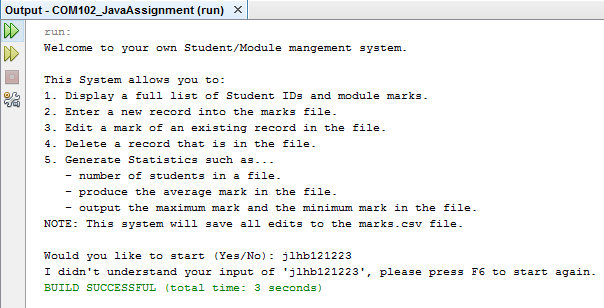
Since they are unexpected, ‘N’ and ‘n’ both fall under the umbrella of valid extreme input. This is because when asked to start, the user is specifically told to enter “Yes” or “No”.





## Test 3: ‘Would you like to start?’ with invalid input

This demonstration is to test whether a user can proceed to use the system after being prompted to enter the expected input but in turn the user enters unexpected, invalid input.

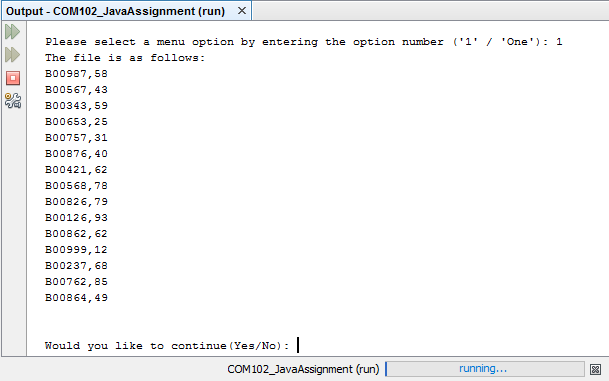


After entering some form of invalid input, the user is told that they have made a mistake, shown what exactly is wrong - “I didn’t understand your input of ‘**(INVALID INPUT)**’ - and shown the shortcut to restart the program in netbeans. Furthermore, the system will end ensuring that they cannot proceed to use the system. See above for examples – invalid inputs are marked in red boxes.

## Test 4: ‘Select a menu option:’ with valid inputs

### Test 4.a: Menu option fullListOfStudents()

This demonstration is to test whether the system proceeds to return the full list of Student IDs with their module marks sourced from the marks.csv file after the user enters the valid and expected input – which is ‘1’.

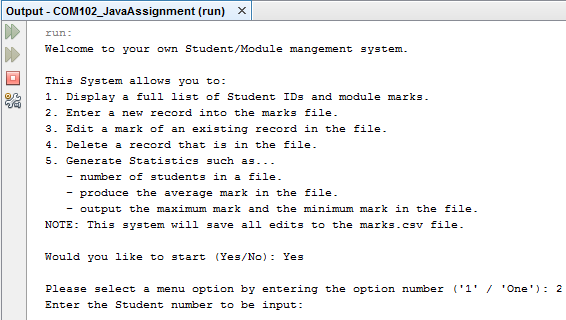


As seen above, when the user enters ‘1’, the system proceeds to run the fulListOfStudents class. Furthermore, once this process has been completed the user will also be asked whether they would like to use the system to perform other tasks - which are featured in the menu.

### Test 4.b: Menu option enterNewRecord()

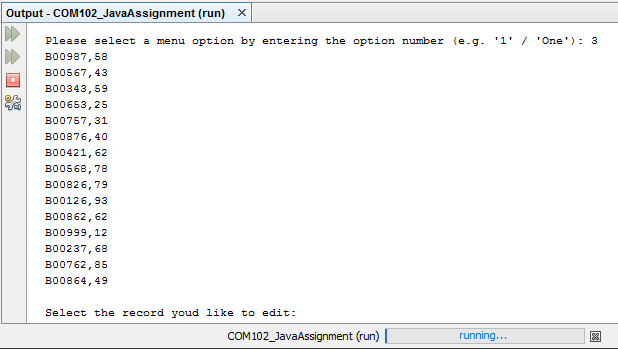
This demonstration is to test whether the system follows the process to allow a new record to be entered into the marks.csv file after the user enters the valid and expected input – which is ‘2’.

We see below that once 2 is entered, the system will proceed to run the enterNewRecord class allowing the user to follow the steps required to add the new record.



### Test 4.c: Menu option editAMark()

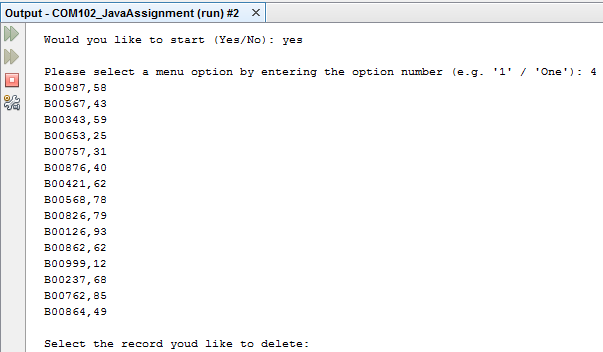
This demonstration is to test whether the system follows the process to allow the user to edit an existing record after the user enters the valid and expected input – which is ‘3’.



As seen above, once the user enters ‘3’, the system follows the proper procedure for the user to edit a record which already exists in the marks.csv file by running the editAMark class.

### Test 4.d: Menu option deleteARecord()

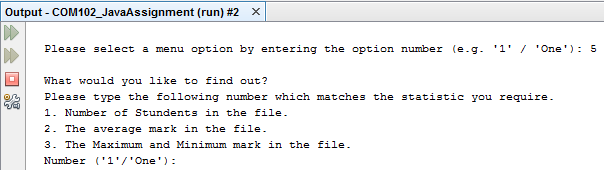
This demonstration is to test whether the system follows the necessary steps to delete an existing record within the marks.csv file after the user enters the valid and expected input – which is ‘4’.



As seen above, when the correct input is entered (4), the program will continue to run the deleteARecord class.

### Test 4.e: Menu option generateStatisticsReport()

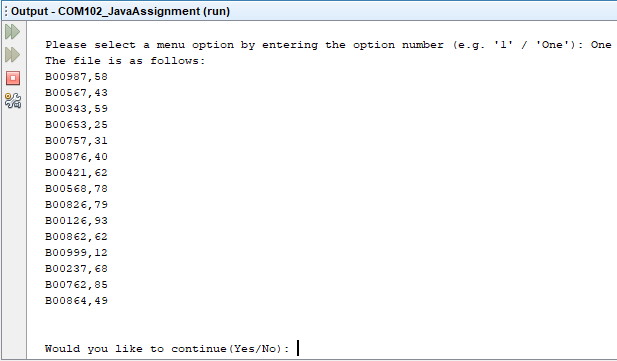
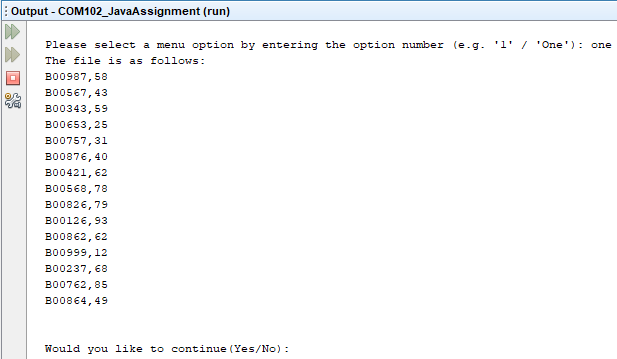
This demonstration is to test whether the system follows the necessary steps to display different statistics based on the marks.csv file, such as: the number of students, the average mark and the minimum and maximum mark. The program will proceed to do this after the user enters the valid and expected input – which is ‘4’ – which is seen below.



## Test 5: ‘Select a menu option:’ with valid extreme inputs.

### Test 5.a: Menu option fullListOfStudents()

This demonstration is to test whether the system proceeds to return the full list of Student IDs with their module marks sourced from the marks.csv file after the user enters the likes of ‘One’ or ‘one’ – which are both examples of inputs which are unexpected but seen as valid.

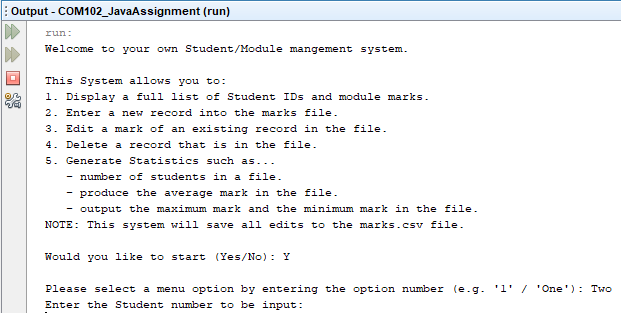
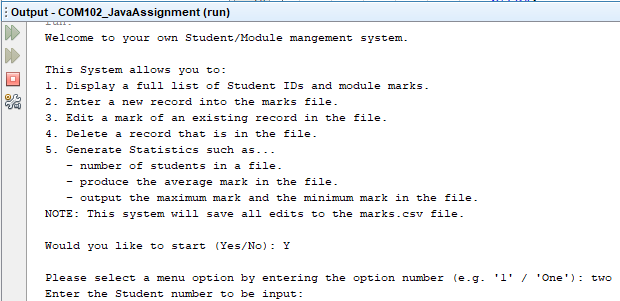


As seen above, when the user enters either ‘one’ or ‘One’, the system proceeds to run the fulListOfStudents class.

### Test 5.b: Menu option enterNewRecord()

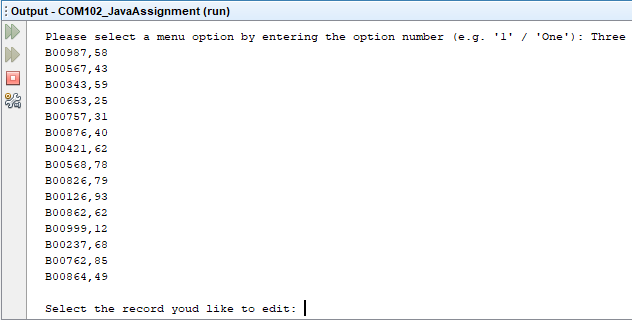
This demonstration is to test whether the system follows the process to allow a new record to be entered into the marks.csv file after the user enters the likes of ‘Two’ or ‘two – which are both examples of inputs which are unexpected but seen as valid.

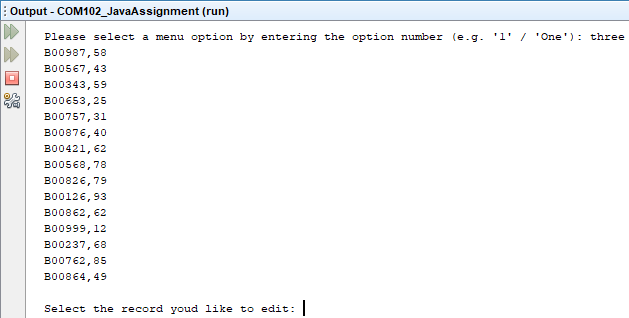
We see below that once either ‘Two’ or ‘two’ are entered, the system will proceed to run the enterNewRecord class allowing the user to follow the steps required to add the new record.



### Test 5.c: Menu option editAMark()

This demonstration is to test whether the system follows the process to allow the user to edit an existing record after the user enters the likes of ‘Three or ‘three – which are both examples of inputs which are unexpected but seen as valid.

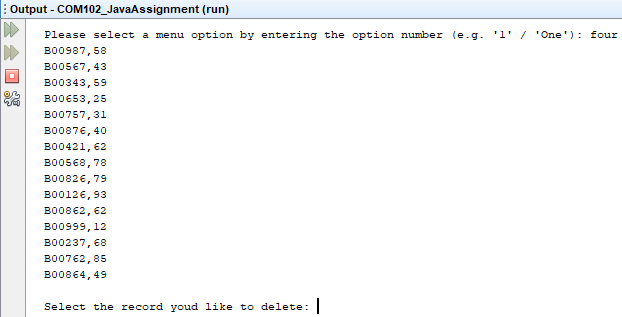
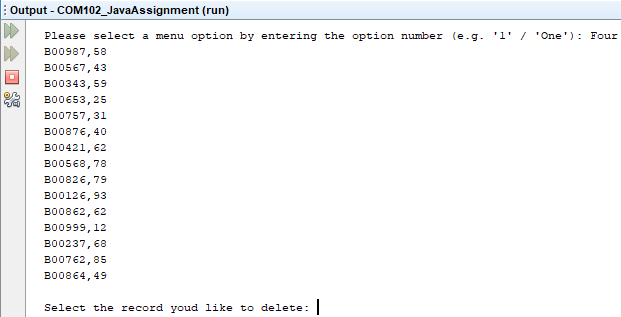




As seen above, once the user enters either ‘three’ or ‘Three’, the system follows the proper procedure for the user to edit a record which already exists in the marks.csv file by running the editAMark class.

### Test 5.d: Menu option deleteARecord()

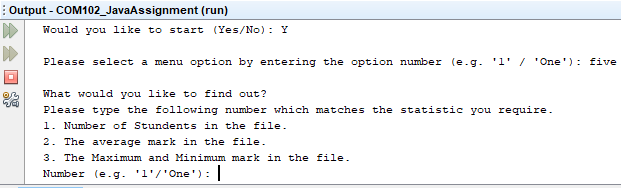
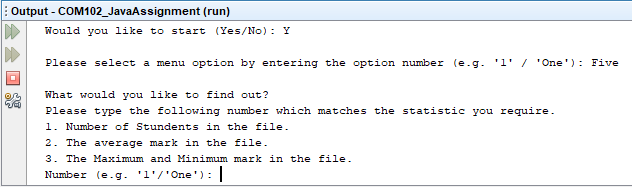
This demonstration is to test whether the system follows the necessary steps to delete an existing record within the marks.csv file after the user enters the likes of ‘Four or ‘four – which are both examples of inputs which are unexpected but seen as valid.



As seen above, when the correct input is entered (‘Four’/’four’), the program will continue to run the deleteARecord class.

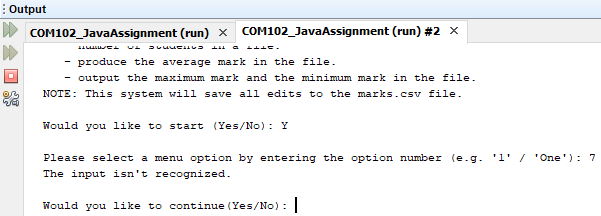
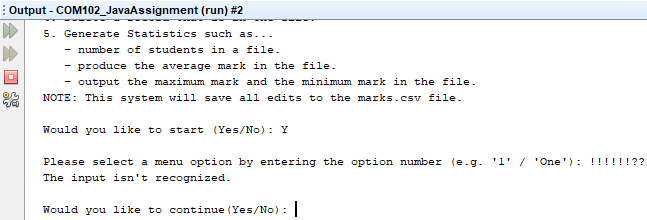
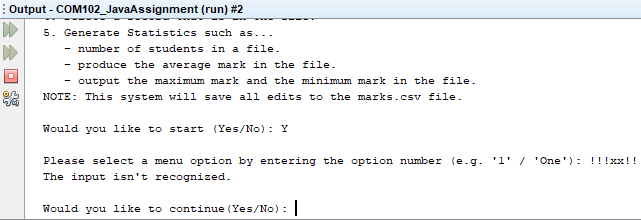
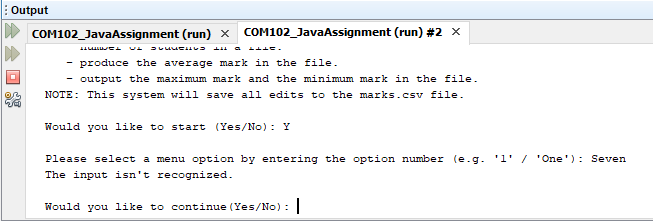
### Test 5.e: Menu option generateStatisticsReport()

This demonstration is to test whether the system follows the necessary steps to display different statistics based on the marks.csv file, such as: the number of students, the average mark and the minimum and maximum mark. The program will proceed to do this after the user enters the likes of ‘Five or ‘five’ – which are both examples of inputs which are unexpected but seen as valid – see below:



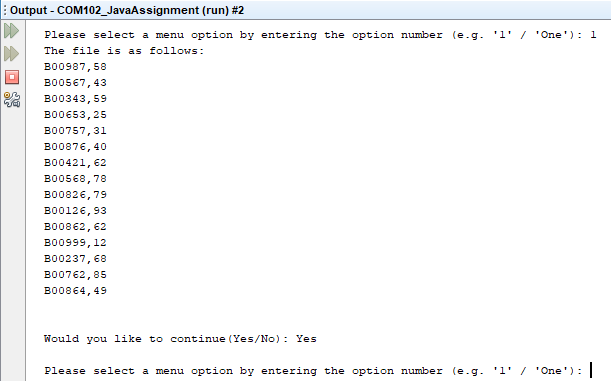
## Test 6: ‘Select a menu option:’ with invalid inputs.

Whether it be characters of the alphanumeric characters or words or digits referring to inexistent menu options, the user will be told they have made a mistake and will be asked whether they would like to continue or not.

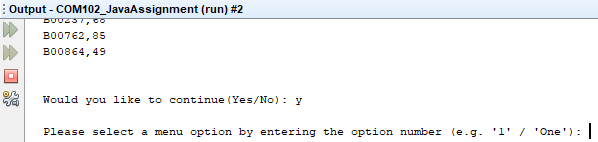
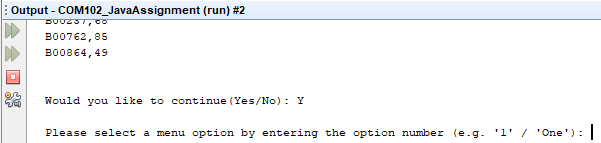
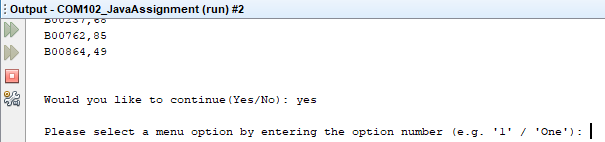


## Test 7: ‘Would you like to continue:’ with valid inputs – ‘Yes’

Once a user exhausts a menu option, he/she will be asked whether they want to continue using the system. The valid inputs to continue to use the system are as follows: ‘Yes’, ‘yes’, ‘Y’ and ‘y’. We see the four valid inputs demonstrated below and after one of the chosen valid inputs are entered, the user will be prompted to choose which menu option they would like to use.

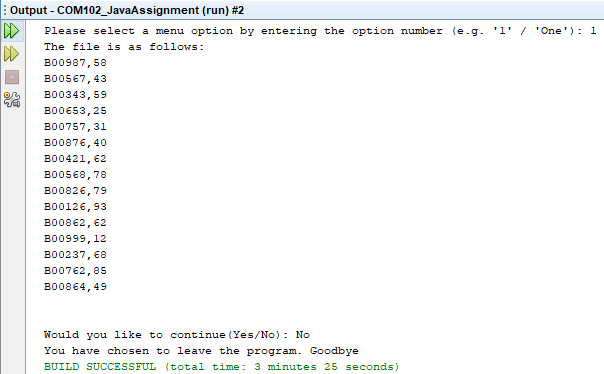


Above is an example of a user exhausting menu option one and then entering one of the valid inputs. In this case, the user has entered ‘Yes’. We see the same below but with the different variations of valid inputs:

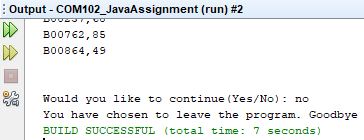
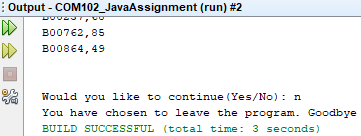
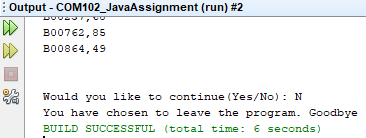


## Test 8: ‘Would you like to continue:’ with valid inputs – ‘No’

Once a user exhausts a menu option, he/she will be asked whether they want to continue using the system. The valid inputs to continue to leave the system are as follows: ‘No, ‘no, ‘N’ and ‘n’. We see the four valid inputs demonstrated below and after one of the chosen valid inputs are entered, the user will be informed that they have chosen to leave the program and the system will stop running.

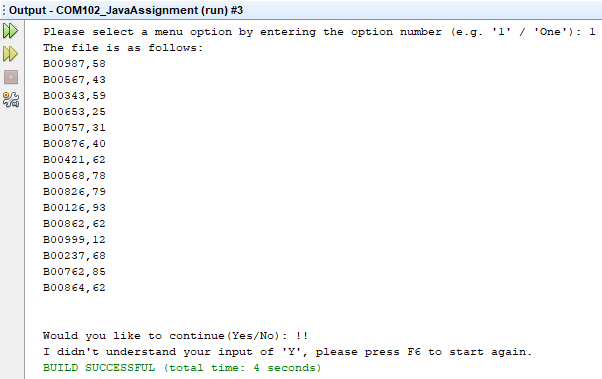


Above is an example of a user exhausting menu option one and then entering one of the valid inputs. In this case, the user has entered ‘No’. We see the same below but with the different variations of valid inputs:



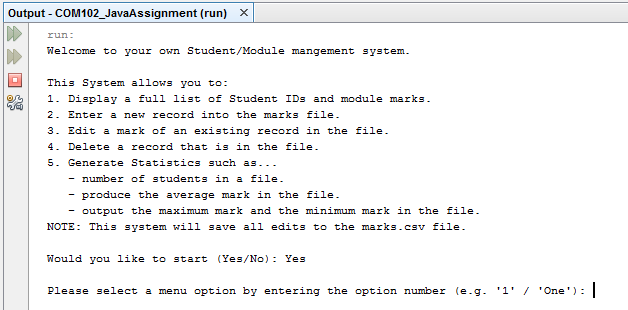
## Test 9: ‘Would you like to continue:’ with invalid inputs

Whether it be characters of the alphanumeric characters or words or digits referring to inexistent menu options, the user will be told they have made a mistake regarding their input, the user will be informed on how to re-run the system through the shortcut of F6 and then the system will exit.

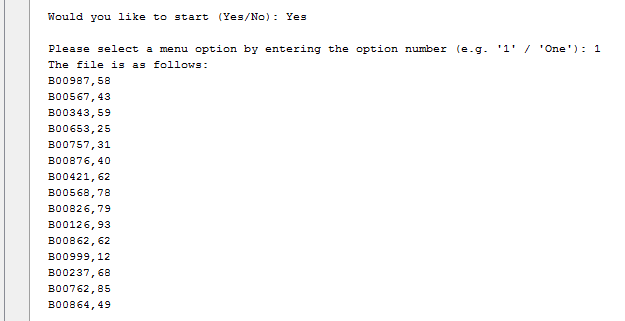


## Test 10: Displaying a full list of correct records

This test demonstrates that once the user has requested the system to return the list of student IDs and Module marks, the system returns this list. The user tells the system they want to proceed and are then asked for what option from the menu they require:



The user can then enter “1” or the word “One” and this will run the class called fullListOfStudents and this will display a full list of records:

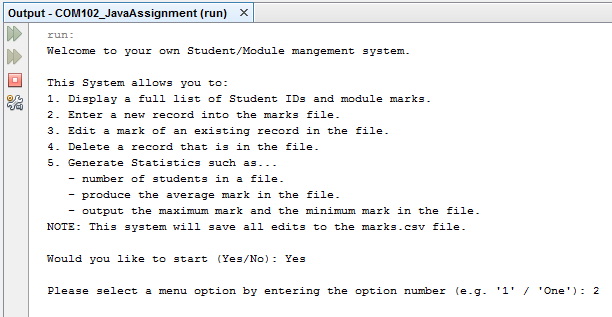


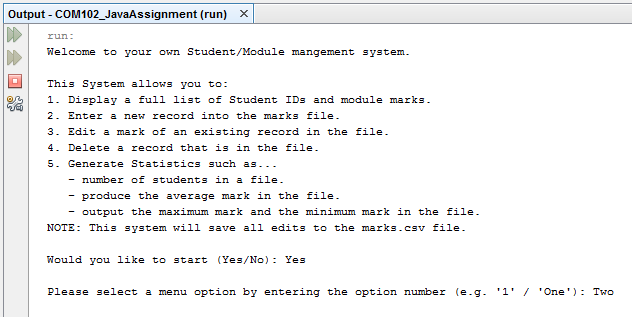
The user is then asked if they would like to continue using the system or exit it:



## Test 11: ‘Enter the student number:’ with valid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system allows them to enter a Student ID and Module mark. The user will enter “2” or “Two” and the system responds accordingly:





The system then asks the user to enter in a new Student ID:



The user can then enter the B Code required, and if correct the system should let the user enter a new student mark:

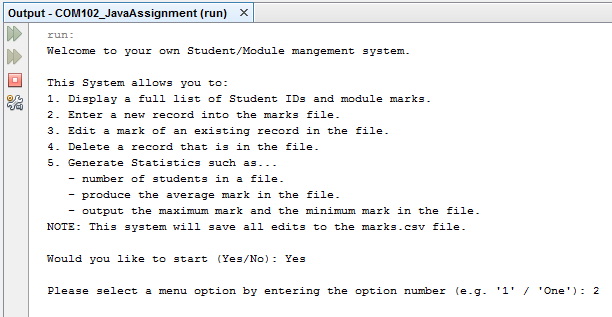


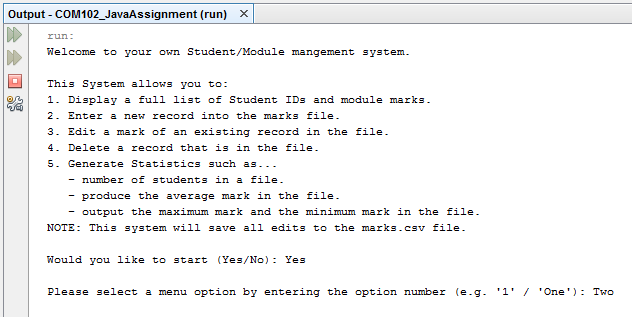
This shows that the system passed this test.

## Test 12: ‘Enter the student number:’ with invalid inputs

### Test 12.a: ‘Enter the student number:’ with invalid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system will not allow them to enter a Student ID and Module mark due to incorrect input. The user has entered “2” or “Two” and the system responds accordingly:

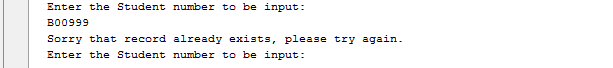




The system then asks the user to enter in a new Student ID, and a B code that already exists is entered:



The system then shows the following error message to the user:

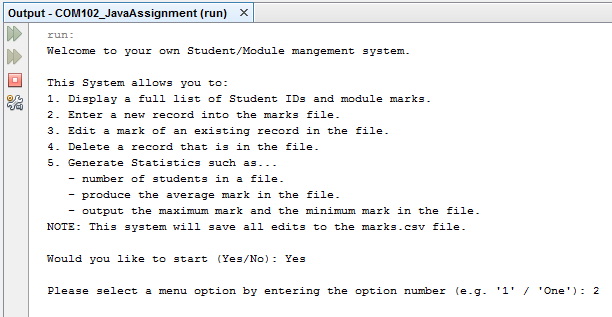


This shows that the system passed this test and lets the user try again.

### 

### Test 12.b: ‘Enter the student number:’ with invalid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system will not allow them to enter a Student ID and Module mark due to incorrect input. The user has entered “2” or “Two” and the system responds accordingly:



The system then asks the user to enter in a new Student ID, the B code is in the incorrect format (There are too many numbers):



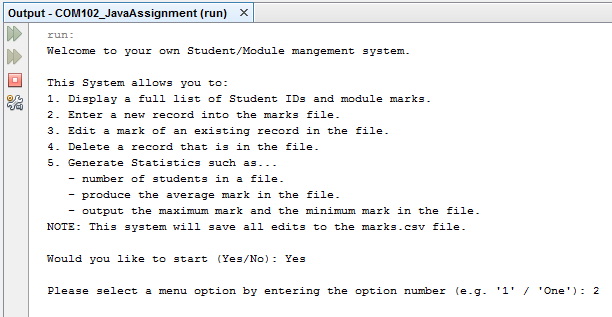
The system then shows the following error message to the user:



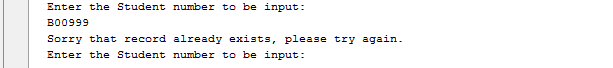
This shows that the system passed this test and lets the user try again.

## Test 13: ‘Incorrect student number, please try again:’ with valid inputs

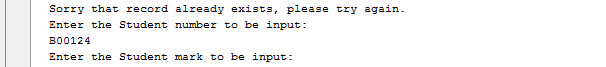
This test demonstrates that once the user has requested the system to add a student to the .csv file, the system will allow them to enter a Student ID again. The user has entered “2” or “Two” and the system responds accordingly:



The wrong B Code is entered, and the system then shows the following error message to the user:



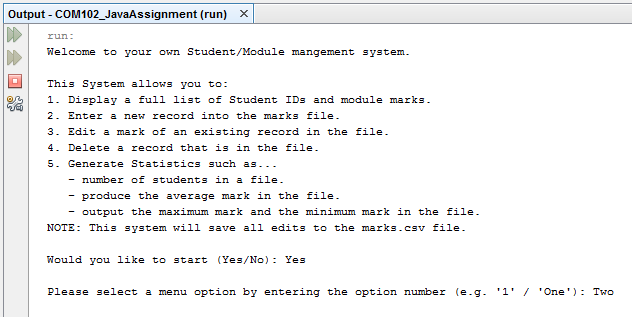
The user can then enter the B Code again, and if correct, the system should let the user enter a new student mark:



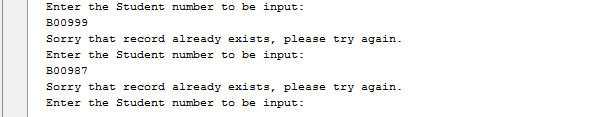
This shows that the system passed this test.

## Test 14: ‘Incorrect student number, please try again:’ with invalid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system will allow them to enter a Student ID again, after failed attempt 2. The user has entered “2” or “Two” and the system responds accordingly:



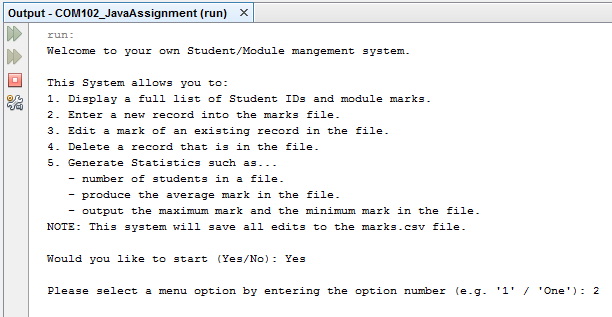
The wrong B Code is entered then another wrong B Code is entered again, and the system shows error message:



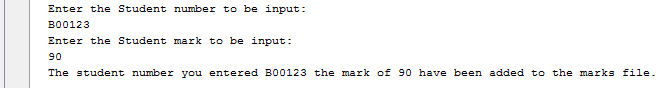
This shows that the system passed this test as the system allows the user to attempt B Code again.

## Test 15: ‘Enter the student mark:’ with valid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system allows them to enter a Student ID and Module mark. The user will enter “2” or “Two” and the system responds accordingly:



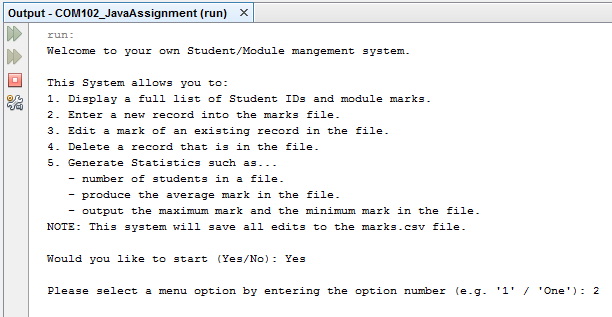
B Code required is entered, and if correct the system should let the user enter a new student mark:



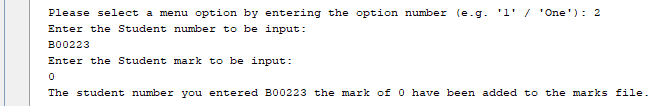
This shows that the system passed this test and confirms that the record has been entered.

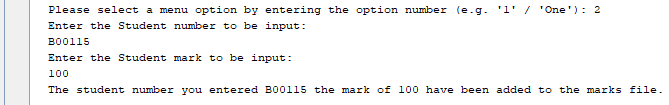
## Test 16: ‘Enter the student mark:’ with valid extreme inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system allows them to enter a Student ID and Module mark. The user will enter “2” or “Two” and the system responds accordingly:



Once a correct B code is correct, the mark is entered, the extreme data in this case is 0 and 100 as this is the lowest and highest values that are still valid

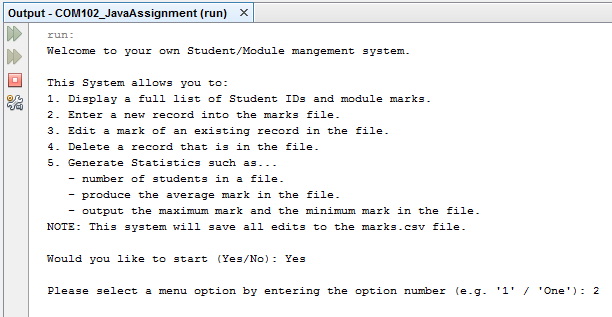




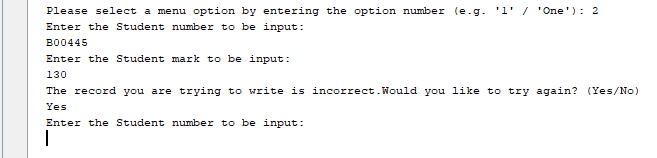
This shows that the system passed this test and confirms that the record has been entered.

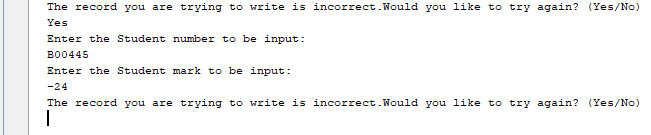
## Test 17: ‘Enter the student mark:’ with invalid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system allows them to enter a Student ID and Module mark, unless invalid data is entered. The user will enter “2” or “Two” and the system responds accordingly:



Once a correct B code is correct, the mark is entered, the extreme data in this case is 0 and 100 as this is the lowest and highest values that are still valid

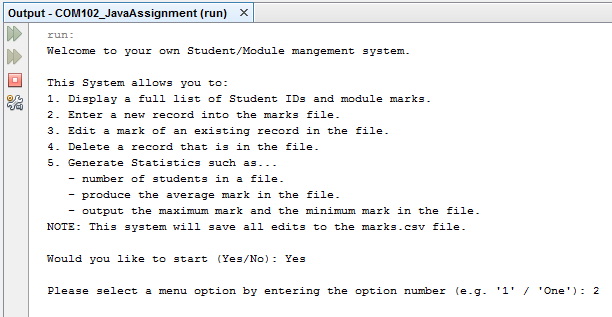




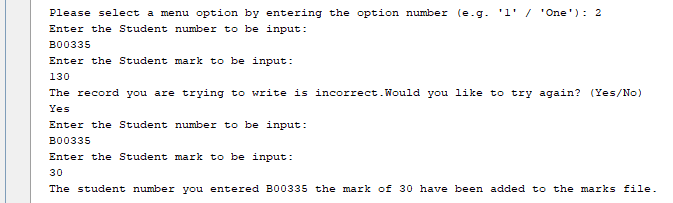
This shows that the system passed this test and confirms that the record has not been entered due to a number over 100 and a number less than 0 being entered.

## Test 18: ‘Invalid data, would you like to try again:’ with valid inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system allows them to enter a Student ID and Module mark, unless invalid data is entered again. The user will enter “2” or “Two” and the system responds accordingly:



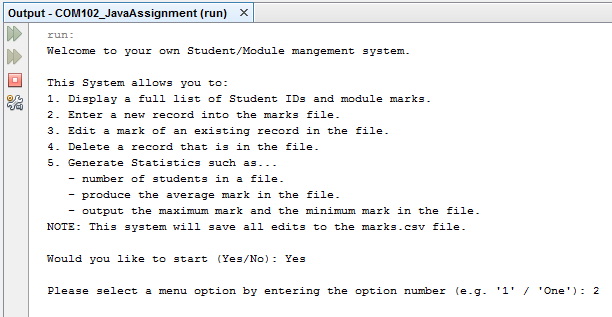
Once a correct B code is correct, the mark is entered, the second attempt will be valid data



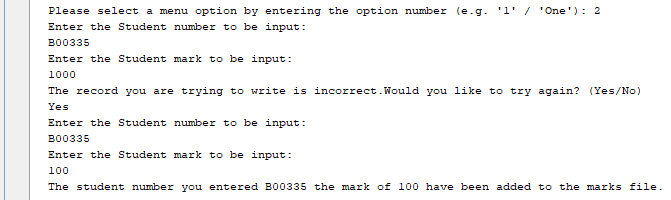
This shows that the system passed this test and confirms that the record has been entered after the second attempt

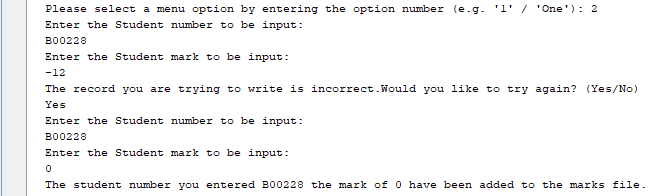
## Test 19: ‘Invalid data, would you like to try again:’ with valid extreme inputs

This test demonstrates that once the user has requested the system to add a student to the .csv file, the system allows them to enter a Student ID and Module mark. The user will enter “2” or “Two” and the system responds accordingly:



Once a correct B code is correct, an incorrect mark is entered, the system then denies and tells the user to try again, the extreme data is entered and in this case is 0 and 100 as this is the lowest and highest values that are still valid

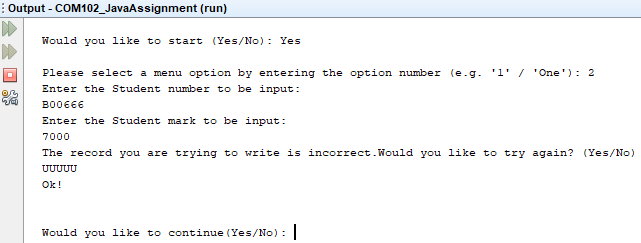




This shows that the system passed the test as when the extreme values are corrected, the records are entered.

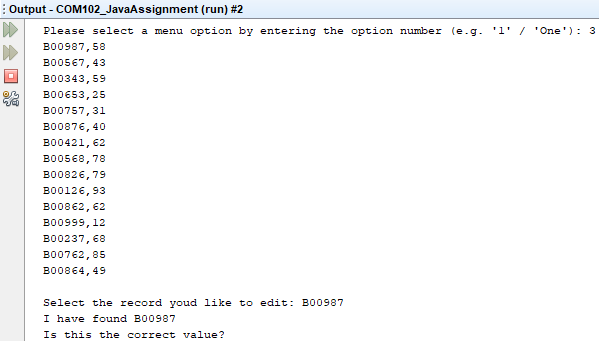
## Test 20: ‘Invalid data, would you like to try again:’ with invalid inputs

If a user enters some type of invalid response to the system informing them that their new record isn’t valid, the system won’t enter these pieces of data into the marks.csv file and will ask the user whether they would like to continue or not.

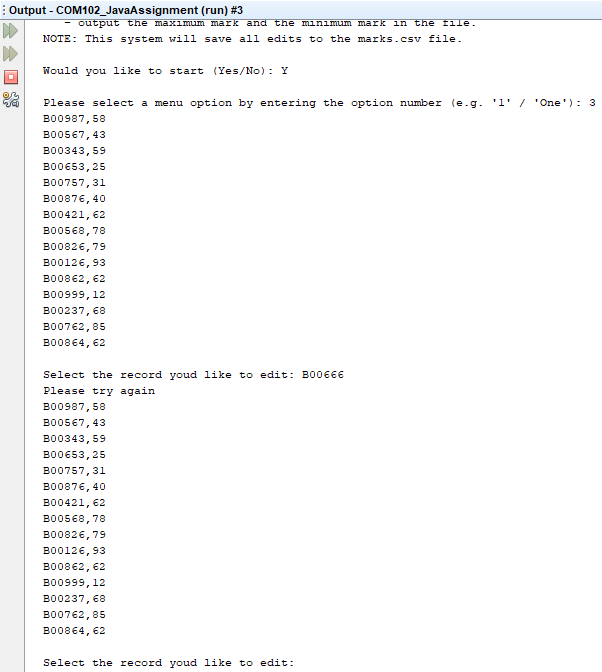


## Test 21: ‘Select the record you would like to edit:’ with valid results

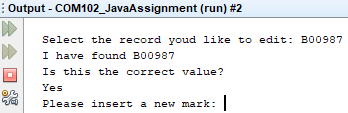
The test below shows that once the user enters an existing record to edit, the system will confirm that the record exists and is able to be edited.



## Test 22: ‘Select the record you would like to edit:’ with invalid results

We see that when the user attempts to edit a record while referring to a Student ID which doesn’t exist, the system will acknowledge this and tells the user that they need to try again.

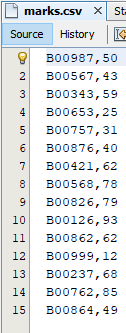
## Test 23: ‘Is that the correct value?’ with valid results

The user can confirm whether the system has collected the proper result to reassure themselves that they aren’t making any mistakes. The valid inputs to confirm are either ‘Yes’ or ‘yes’ as shown below.

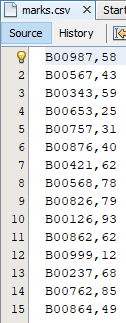
## Test 24: ‘Insert a new mark:’ with valid results

Once the user has confirmed that the system has found the proper Student ID and he/she is happy to continue, they are prompted to enter a new mark. When we consider what a valid mark is, it is one which is in the range of the digits 0 and 100.

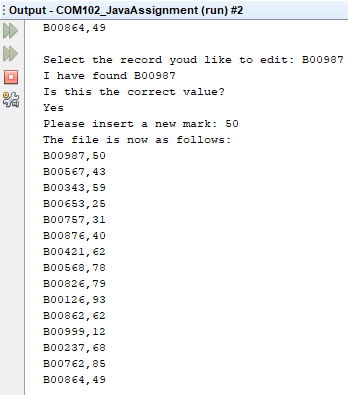
Example of a valid input being entered and proof that the new mark is updated in the marks.csv file.



Updated file with new mark



Original file with original mark



Updated records returned

User’s input

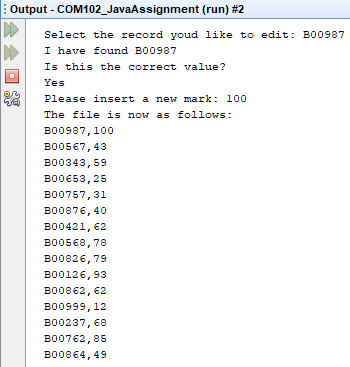
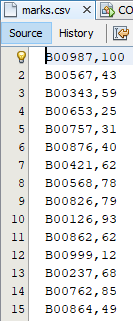
Updated mark

## Test 25: ‘Insert a new mark:’ with valid extreme results

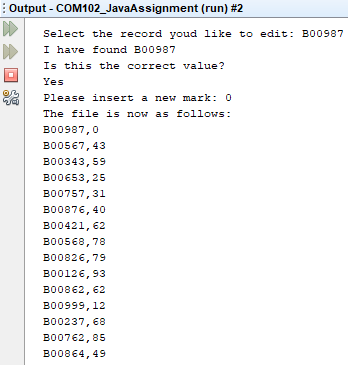
Once the user has confirmed that the system has found the proper Student ID and he/she is happy to continue, they are prompted to enter a new mark. When we consider what a valid extreme mark is in this case, it is one which is either the digit 0 or 100.

Example of both valid extreme inputs being entered and proof that the new mark is updated in the marks.csv file.

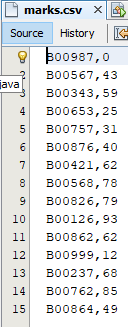
Updated file with new mark



Updated mark



Updated mark

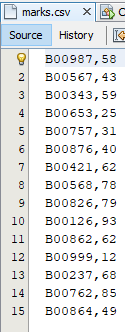
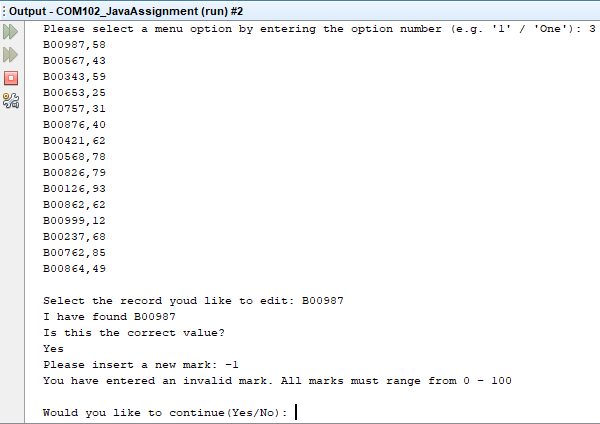


Updated file with new mark

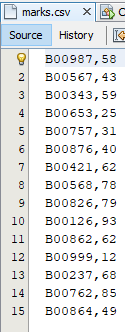
## Test 26: ‘Insert a new mark:’ with invalid results

The system does not allow the user to enter a new mark anywhere below 0 or above 100. If a user tries to enter such marks, they will be informed that they have entered an invalid mark, told that they can only enter mark values in the range of 0-100 and are asked if they would like to continue to use the system. Furthermore, the marks.csv file will not update their chosen record with the invalid mark.

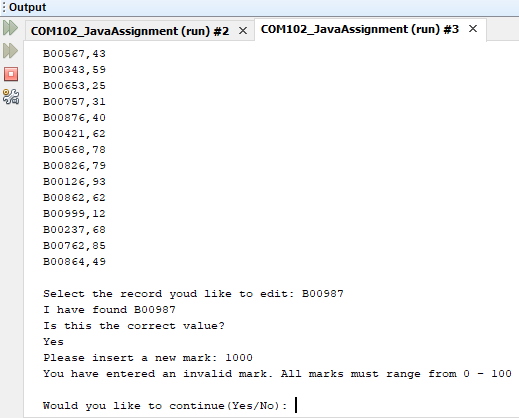
We see examples below:



Mark stays the same

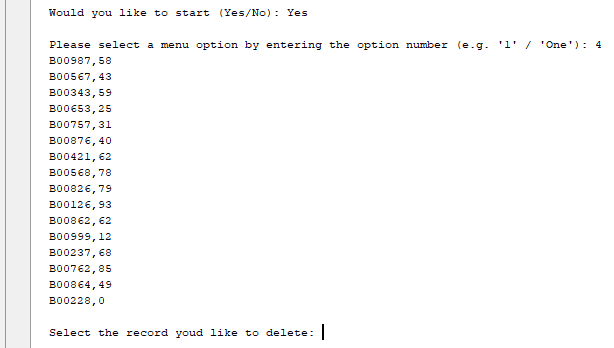


Mark stays the same



## Test 27: ‘Select the record you would like to Delete:’ with valid results

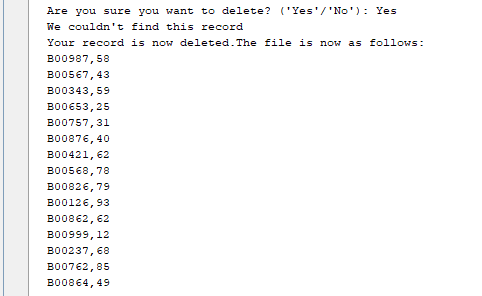
This test shows that the system can remove records from .csv files when requested:



The user then simply enters the b Code to be deleted and then asks for confirmation:

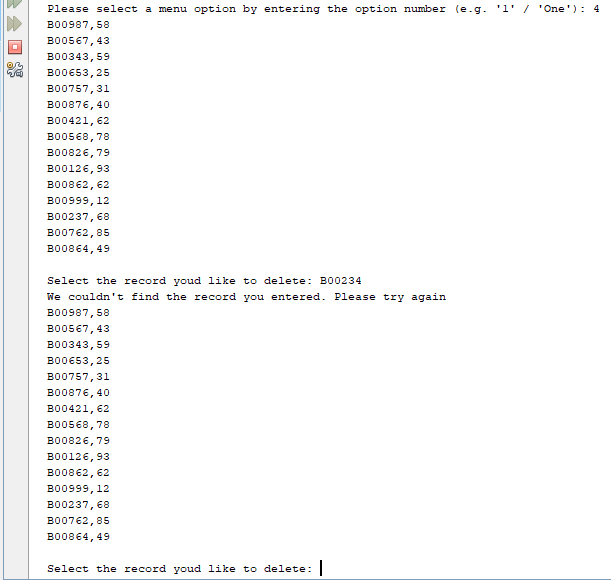


If the user enters yes, the record is deleted, if not the system ignores the request:



This shows that the system has passed this test because the record is no longer there

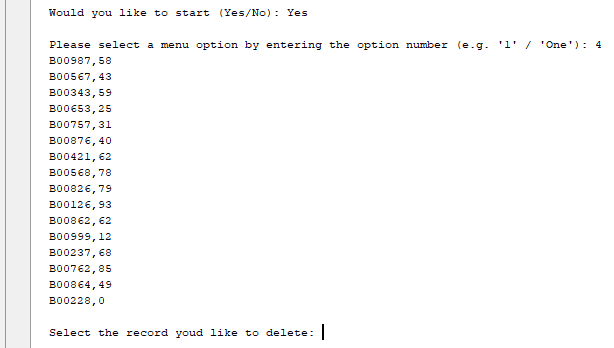
## Test 28: ‘Select the record you would like to delete:’ with invalid results

This test shows that if the user enters a B Code that doesn’t exist, the system will not let it be removed:

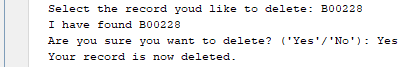
This system passes this test as it will not let the user remove a record that doesn’t exists and let them try again.

## Test 29: ‘Is that the correct value?’ with valid results

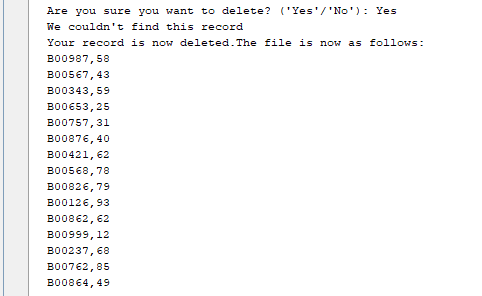
This test shows that the system can remove records from .csv files when requested:



The user then simply enters the b Code to be deleted and then asks for confirmation:



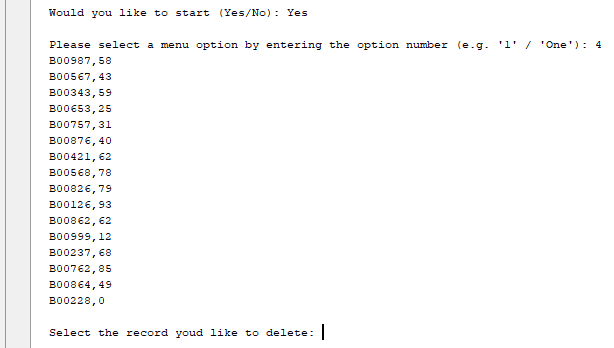
If the user enters yes, the record is deleted, if not the system ignores the request:



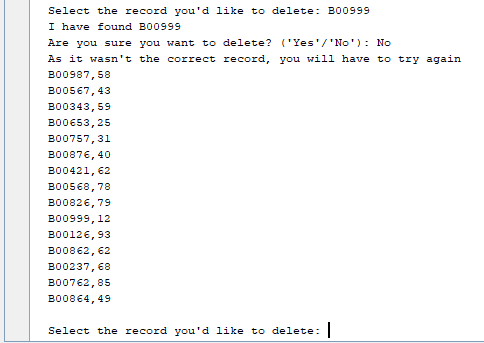
This shows that the system has passed this test because the record is no longer there

## Test 30: ‘Is that the correct value?’ with invalid results

This test shows that the system can remove records from .csv files when requested:



The user then simply enters the b Code to be deleted and then asks for confirmation:

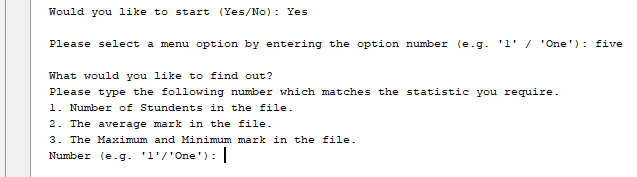


This shows that the system has passed this test because the system allows the user to try again.

## Test 31: ‘Select a menu option:’ with valid inputs

### Test 31.a: Menu option numOfRecords ()

This test shows that when the system user requests a statistic report the system returns a secondary menu, and lets the user choose the statistic they need:



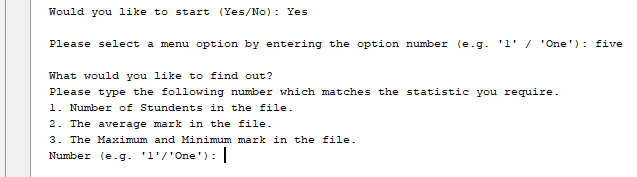
The user then requests option 1 by typing in the number 1 or the word:



This test shows that the system has passed and shows the number of records in the file

### Test 31.b: Menu option averageMark ()

This test shows that when the system user requests a statistic report the system returns a secondary menu, and lets the user choose the statistic they need:



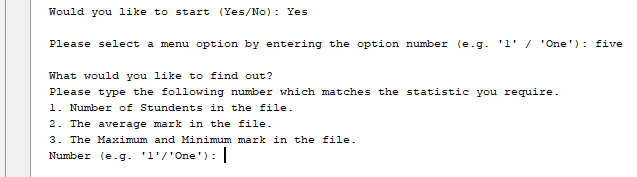
The user then requests option 1 by typing in the number 1 or the word:



This test shows that the system has passed and shows the number of records in the file

### Test 31.c: Menu option minAndMax ()

This test shows that when the system user requests a statistic report the system returns a secondary menu, and lets the user choose the statistic they need:



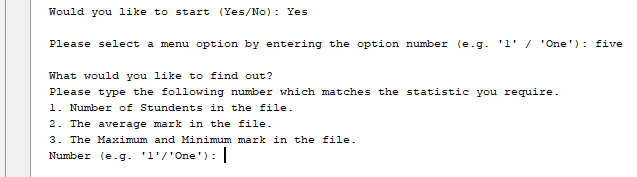
The user then requests option 1 by typing in the number 1 or the word:



This test shows that the system has passed and shows the number of records in the file

## Test 32: ‘Select a menu option:’ with valid extreme inputs.

This test shows that when the system user requests a statistic report the system returns a secondary menu, and lets the user choose the statistic they need:



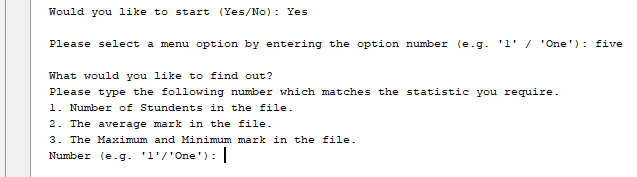
The user then requests option 1 by typing in the word as this would be an extreme input



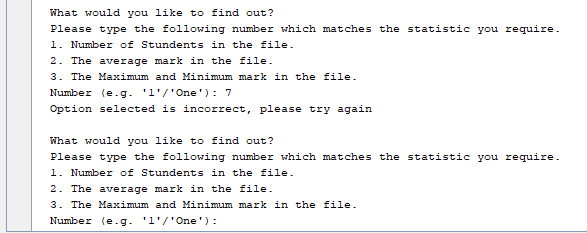
This test shows that the system has passed and shows the number of records in the file

## Test 33: ‘Select a menu option:’ with invalid inputs.

This test shows that when the system user requests a statistic report the system returns a secondary menu, and lets the user choose the statistic they need:



The user then requests option 1 by typing in the word as this would be an extreme input



This test shows that the system has passed and allows the user to retry their option.