

Software Development 2

Assignment 2

Programmer: *Sarunas Kucys*

Student Number: *B00100612*

Assignment: *Grade Analyzing Program*

Written: *10.03.2018*

Contents

[Introduction 3](#_Toc508442441)

[Description of program design 3](#_Toc508442442)

[Testing 6](#_Toc508442443)

[Test table 6](#_Toc508442444)

[Test 1 7](#_Toc508442445)

[Test 2 8](#_Toc508442446)

[Test 3 9](#_Toc508442447)

[Test 4 10](#_Toc508442448)

[Test 5 11](#_Toc508442449)

[Test 6 12](#_Toc508442450)

[Test 7 13](#_Toc508442451)

[Test 8 14](#_Toc508442452)

[Test 9 15](#_Toc508442453)

# Introduction

This Software Development assignment is based on developing a simple grade analyzing program for analyzing class tests. It is (in theory) created for ITB lecturers and provides possibility to gather and analyze information of small class student grades. The program is developed for showing knowledge of how to use arrays, process arrays, search and sort arrays. Also - implementing methods for calculating and showing the lowest, highest and average grades, searching and sorting arrays. This program is simple, but at the same time - quite useful for analyzing purposes.

# Description of program design

Successful creation of program such as this one requires more than correctly working code. In my honest opinion, good programming practices should be applied all the time with software developing, regardless of what purpose a program serves. This might be often overlooked, but I think any programmer should make their code look neat and clean any time when developing software.

Before the beginning of designing this program, I’ve decided to start with writing correct methods, as they will be called later on. I’ve put them at the end of the program code and for less confusion. They were one after another in order to match selection menu, which means if first menu selection is to show lowest grade and it calls corresponding minimum method, then same method is found first after main method. This might seem not important, but in fact it helps troubleshoot problems (at least for myself) if any encountered.

When designing this program, one of the most important things was permitting no entry of erroneous data. Solution here was to use while loops. For example, before the switch statement I decided to check user input for menu selection option:

**while (optionSelection<=0 || optionSelection>6) //only valid selection.**

**{**

**System.out.print("Please enter a valid option (1-6): "); //error message**

**optionSelection = userInput.nextInt(); //prompt to re-enter**

**}**

For loop was used for collecting names and grades and passing them into corresponding nested arrays. First I tried creating two for loops that would make user to fill up names array first and then grades array separately, which in this case is inconvenient. The solution in this bit was to use one for loop instead of two, so user can enter a grade of corresponding student right after entering a name.

**for (int i = 0; i < students.length; i++)**

**{**

**System.out.print("Please enter name of student " + (i+1) + ": ");**

**students[i] = userInput.next();**

**System.out.print("Please enter the grade for " + students[i] + ": ");**

**grades[i] = userInput.nextDouble();**

**System.out.println("");**

**}**

For the menu options - switch statement was chosen for its simplicity and for elimination of need to use if/else statement, which is slightly longer and might be little more confusing.Each option in the menu is matching a case in the switch statement, where methods are called after user selection.

Another problem to address was menu re-usability. If a user wants to analyze not one but few things, the menu must work again after selection is made. This was solved by using do/while loop:

**do //starts the loop**

**{**

**optionSelection = userInput.nextInt(); //allows user to select an option from the menu**

**…**

**}**

**while (optionSelection == 1 || optionSelection == 2 || optionSelection == 3 || optionSelection == 4 || optionSelection == 5 || optionSelection == 6);**

# Testing

## Test table

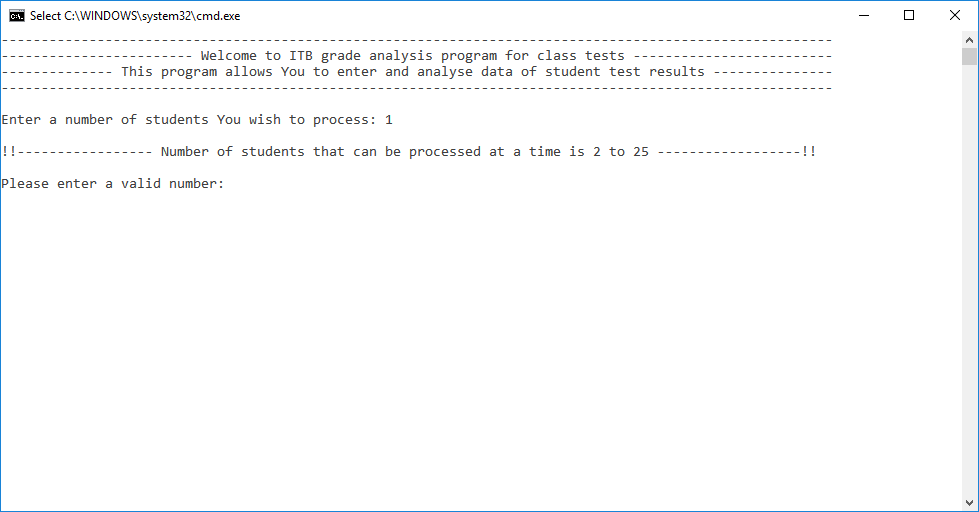
|  |  |  |
| --- | --- | --- |
| Test No. | Description | Sample Test Data |
| 1 | Testing a while loop by entering a number of students different than 2 to 25. | 1 |
| 2 | Testing if for loop is working correctly with two nested arrays | John, 12 |
| 3 | Testing a while loop by entering a grade different than 1 to 100. | 200 |
| 4 | Testing a minimum method. | 12, 35, 65, 87, 91, 99 |
| 5 | Testing a maximum method. | 12, 35, 65, 87, 91, 99 |
| 6 | Testing an average method. | 12, 35, 65, 87, 91, 99 |
| 7 | Testing grades sorting method. | 12, 35, 65, 87, 91, 99 |
| 8 | Testing linear search method. | John |
| 9 | Testing do/while loop. | Menu options 1-5 |

### Test 1

Enter 1 for number of students.

Expected result: Error message appears. Re-typing needed.

Result is as expected:



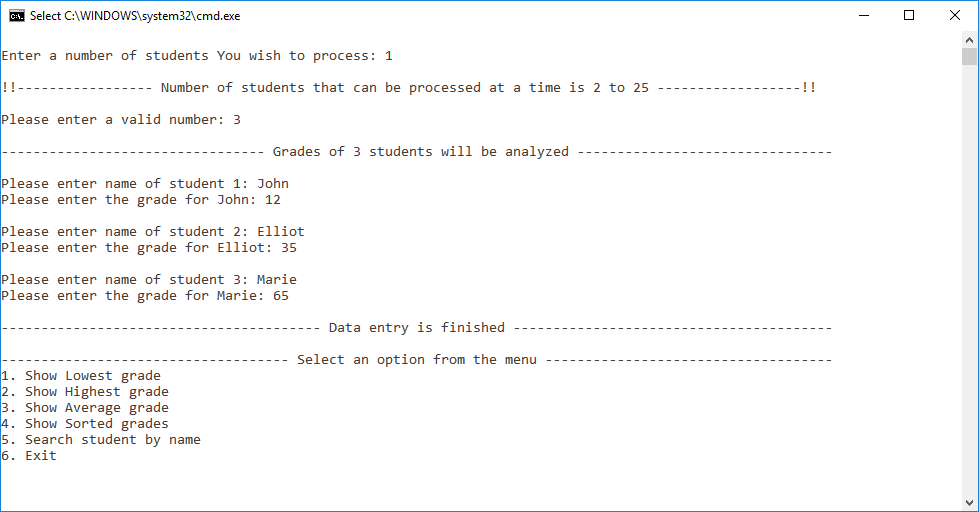
2

### Test 2

Select 3 students to process, enter name John and grade 12, Elliot and grade 35, Marie and grade 65.

Expected result: User is prompted to enter name of student 1 and grade for student 1, then user is prompted to enter name of student 2 and grade for student 2 and so on…

Result is as expected:

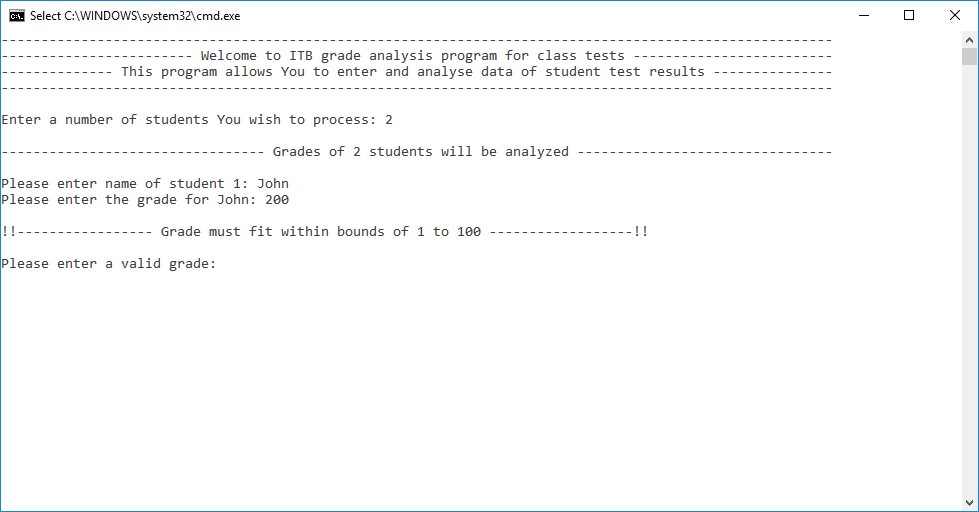


### Test 3

Enter 200 for grade of student 1.

Expected result: Error message appears. Re-typing needed.

Result is as expected:

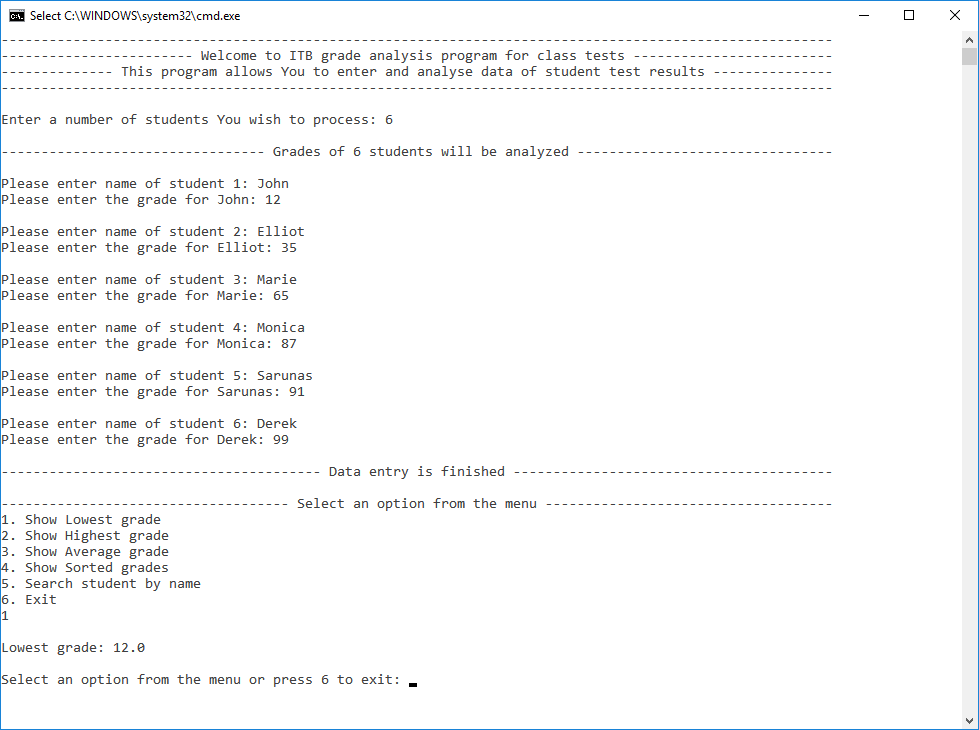


### Test 4

Testing if minimum method is working correctly.

Expected result: After entering names and grades, minimum method calculates and returns the lowest grade.

Result is as expected:

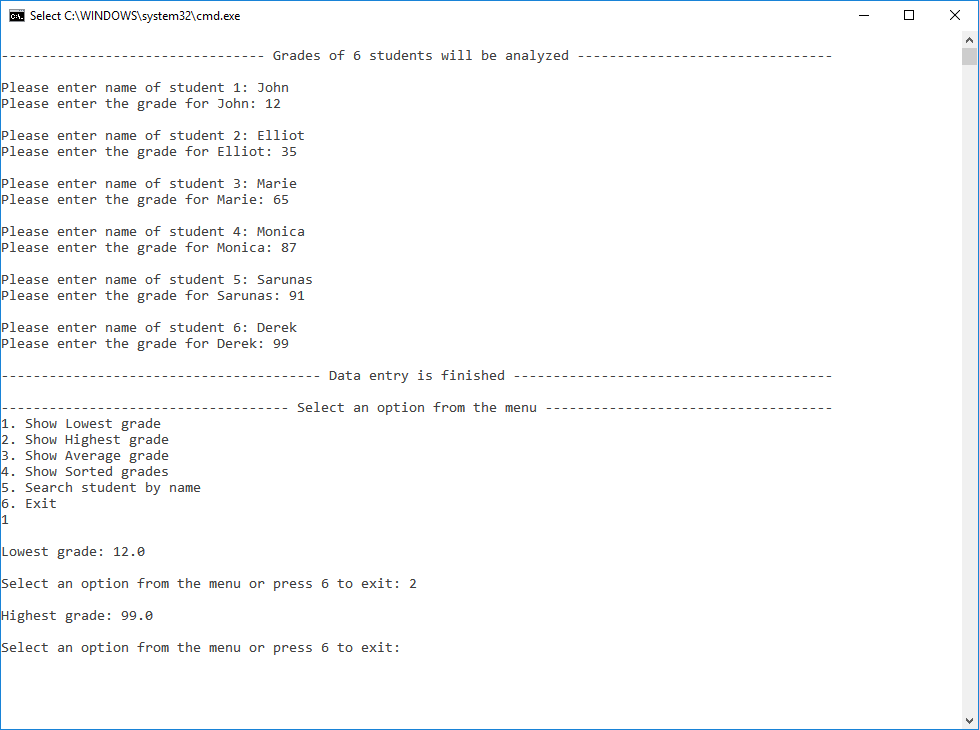


### Test 5

Testing if maximum method is working correctly.

Expected result: After entering names and grades, maximum method calculates and returns the highest grade.

Result is as expected:

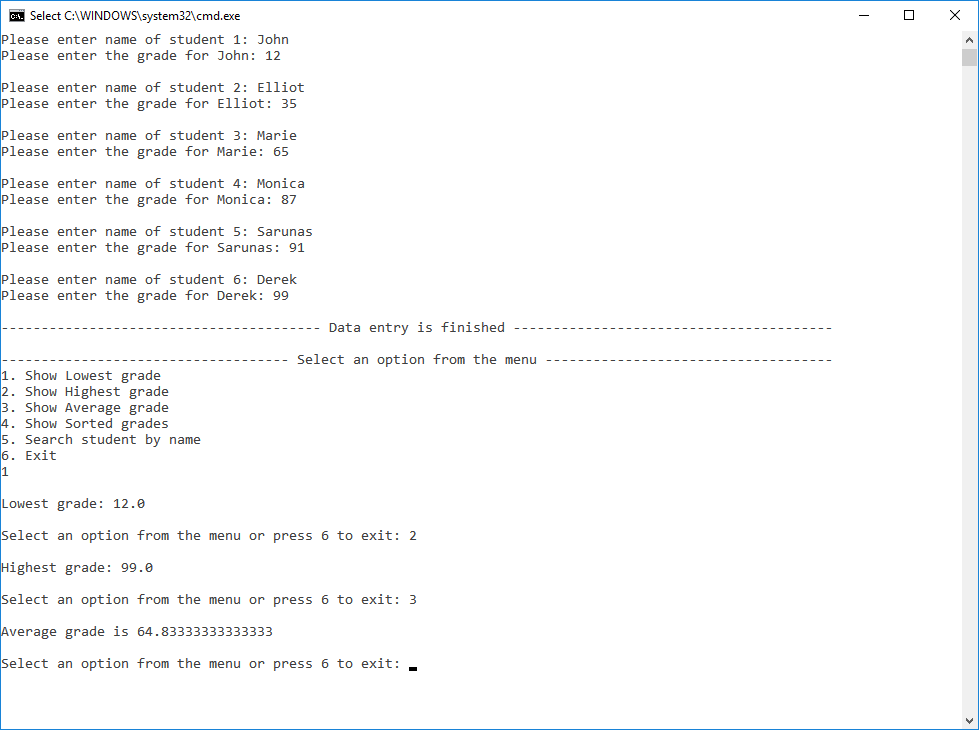


### Test 6

Testing if average method is working correctly.

Expected result: After entering names and grades, average method calculates and returns an average grade.

Result is as expected:

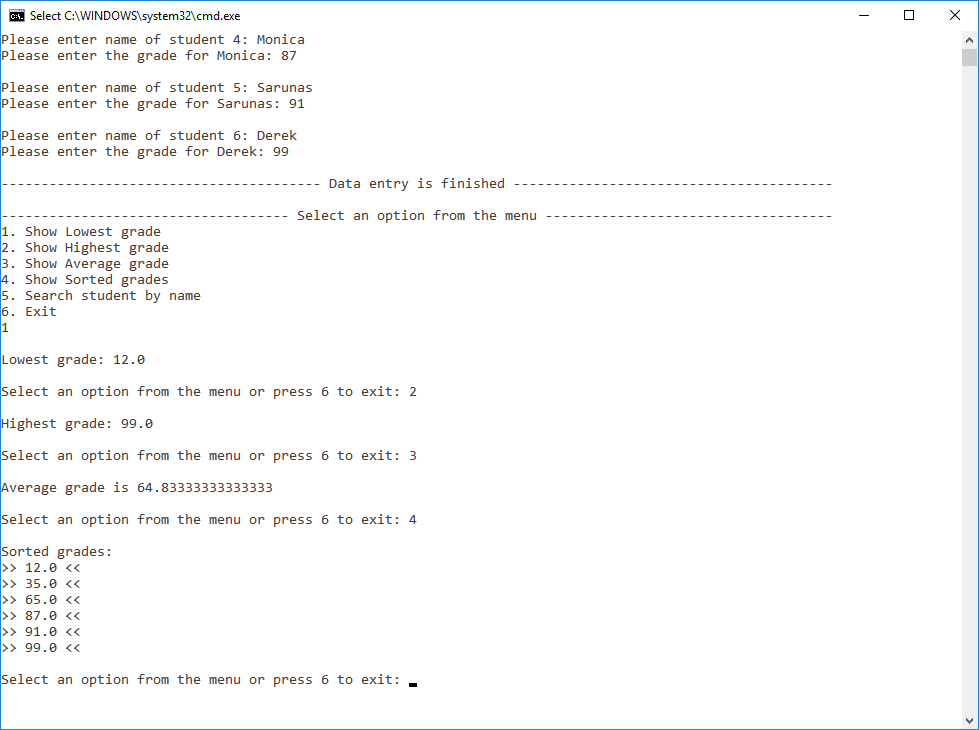


### Test 7

Testing if grades sorting method is working correctly.

Expected result: After entering names and grades, grade sorting method shows grades lowest to highest.

Result is as expected:

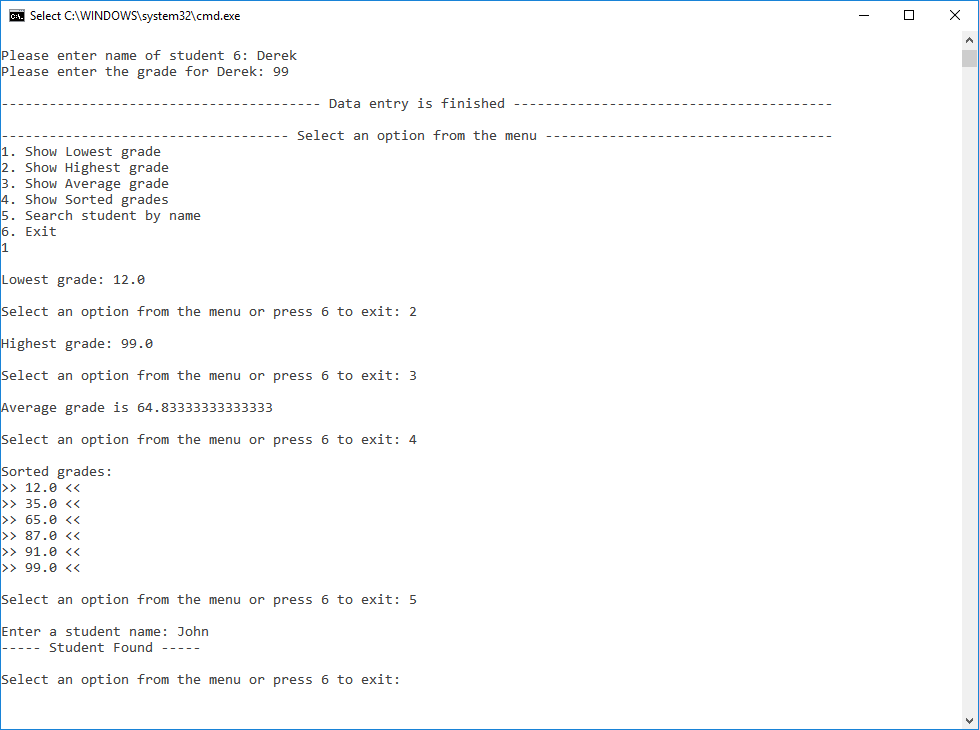


### Test 8

Testing if linear search method is working correctly.

Expected result: After entering names and grades menu option 5 is selected and name John is entered. Linear search method finds a name in students array and message “Student Found” appears.

Result is as expected:



### Test 9

Testing if do/while loop is working correctly.

Expected result: After entering names and grades menu option 1 to 5 is selected. Program then does not terminate, but allows user to select another option from the menu.

Result is as expected:

