Classroom

## Problem definition

Schoolteachers encounter a large amount of work at the end of the scholastic year, when they need to keep a record of marks gained by their students and assign grades correctly. Accuracy is important, and it is also essential to ensure that the correct grades are awarded. All this needs to be completed quickly, so that results can be sent to the students before the end of the year.

## Task definition

The *Classroom* program is meant to solve these problems by providing a solution that stores the students’ marks, assigns grades and outputs them on screen. It also calculates and displays the total, average, highest and lowest marks, as well as the number of students that obtained marks above and below average. Details of specific students can be searched, using the student’s name.

All these functions can be accessed via a menu that is easy to use and understand. Input is done through the keyboard. The interface is text-based and can be accessed via the command prompt or a terminal window.

### Inputs

* Password

To access the program, the user needs to input a numeric password. This is a special feature to ensure privacy of the students’ marks.

* Number of students

After obtaining access, the user needs to enter the number of students in the class. This value will be used throughout the program.

* Student details

The name and mark (out of 100) of each student needs to be input, using the appropriate menu option.

* Menu option

To access the different functions, the user needs to input the relative menu option.

### Outputs

* Student details

Each student’s properties, including their marks and grades, are displayed in sequential order.

* Total marks

The sum of all the marks obtained is displayed on screen.

* Average mark

The average mark within the classroom is displayed on screen.

* Highest mark

The highest mark obtained is displayed on screen.

* Lowest mark

The lowest mark obtained is displayed on screen.

* Students above average mark

The number of students that obtained a mark higher than the average mark is displayed on screen.

* Students below average mark

The number of students that obtained a mark lower than the average mark is displayed on screen.

## Algorithm

This program is made up of four classes, including the Keyboard class that is used to control user input via the keyboard. The other three classes are listed and explained in this section.

### StudentMarks

This is the main class of this program, containing a single method, which is the ‘main’ method. This method controls access to the program via password input and checking, displays the menu and executes the options requested by the user.

### Student

The student class is used to enable objects of students to be created and stored by the program. This class provides the following properties for each student object:

* Student Name
* Mark
* Grade

This class can be instantiated via two different constructors. A default constructor creates an object with no values set to it. A second constructor expects the student name and mark as input parameters. The student’s grade is calculated during instantiation.

### Classroom

The classroom class hosts the data on all the students in an array containing objects of the student class. This array is then used in different methods, that carry out tasks requested by the user via the menu. These methods include:

* getTotalMark

This method loops through the array of students and returns the total of their marks.

* getAverageMark

This method uses the getTotalMark method to calculate the average mark of the students, by dividing it by the number of elements in the array of students.

* getStudentsAboveAverage

This method uses the getAverageMark method when looping through the array of students to output the number of students who obtained a mark greater than the average mark.

* getStudentsBelowAverage

This method uses the getAverageMark method when looping through the array of students to output the number of students who obtained a mark that is less than the average mark.

* getHighestMark

This method loops through the array of students to identify and outputs the one with the highest mark.

* getLowestMark

This method loops through the array of students to identify and outputs the one with the lowest mark.

* getFirstInClass

This method returns the name of the student identified in the getHighestMark method.

* searchStudent

This method enables the user to search through the array of students, using the student’s name. It returns the Student object found. If no student is found with a name that matches the criteria, a Student object with a name of “Not Found’ is returned.

* getGradesIndex

This method returns an index representing the grade obtained. It is a private method, which is used internally by the Classroom class to generate the grades histogram.

* getMarksIndex

This method returns an index representing the mark obtained. This is done by dividing the mark by 10 and converting the result to an integer value. It is a private method, which is used internally by the Classroom class to generate the marks histogram.

* getStudentsByGrade

This method returns an integer array with the number of students at each grade index, referred to in the getGradesIndex method. This private method is used to generate the grades histogram.

* getStudentsByGrade

This method returns an integer array with the number of students at each mark index, referred to in the getMarksIndex method. This private method is used to generate the marks histogram.

* printHistogram

This method used the values provided by the getStudentsByMark and getStudentsbyGrade methods to display a histogram by grades or by mark, according to what the user has requested.

## Special features

* Password

To access the program, the user must enter a password, made up of a sequence of numbers. This password is hard-coded and cannot be modified.

* Menu

All the features of this program can be accessed via a menu system, which explains the options available and how to access them.

* Test data generation

There is a hidden feature in the program that generates test data, including random marks for each student in the classroom. This can be used both for testing purposes and also for demonstrating the program’s capabilities. To access this feature, the user needs to input ‘99’ at the menu prompt.

* Visual histograms

The program is able to generate and display two types of histograms – one based on the different grades (A, B, C, F and U), and another one based on categories of ranges of ten marks (0 – 9, 10 – 19, etc.). These histograms present the user with a visual representation of the spread of marks or grades, and therefore provide an easy-to-use display of the students’ performances.