Programming I Project 1 Documentation

Introduction: first draft

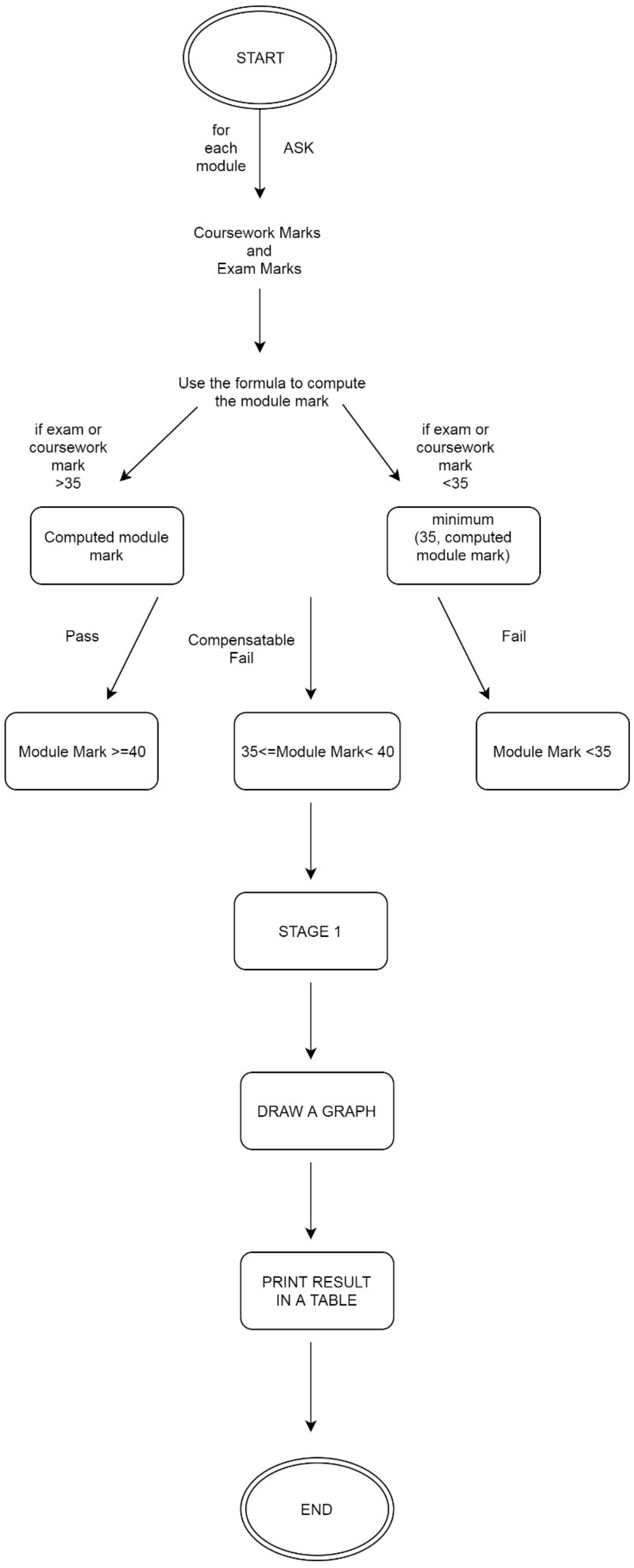
In computer science, the reason for creating programs is to give users useful and easy to use tools in order to allow them to find the faster way to get what they need.

While creating my program I came through difficulties and challenges.

In this documentation, I will outline the process I went through to solve these problems.

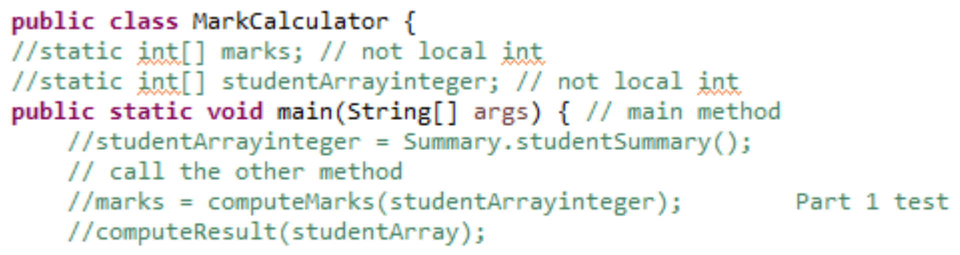
At first, I encountered a problem while organizing the structure of the program due to the fact that we had an enormous freedom in interpreting the text and creating and organizing the tasks.

So I tried to draw a scheme of how the program should have been designed.


START
for each module
ASK
Coursework Marks and Exam Marks
Use the formula to compute the module mark
if exam or coursework mark >35
if exam or coursework mark <35
Computed module mark
minimum (35, computed module mark)
Pass
Compensatable Fail
Fail
Module Mark >=40
35<=Module Mark< 40
Module Mark <35
STAGE 1
DRAW A GRAPH
PRINT RESULT IN A TABLE
END

Purpose of the program

After a first draft, I started to write some lines for the software that elaborate Stage 1 student marks. It is still possible to read an early version of the code for Part 1 in the MarkCalculator class. The tasks were divided into two parts: part one that gave the bases of the program and a second part was more focused on how to improve the user's experience.



```
public class MarkCalculator {
//static int[] marks; // not local int
//static int[] studentArrayinteger; // not local int
public static void main(String[] args) { // main method
    //studentArrayinteger = Summary.studentSummary();
    // call the other method
    //marks = computeMarks(studentArrayinteger);        Part 1 test
    //computeResult(studentArray);
```

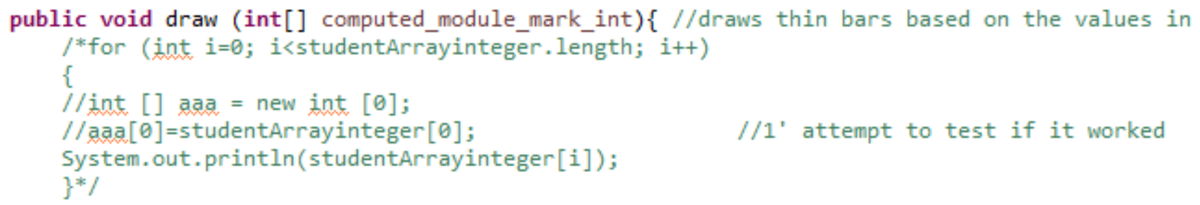
In this class, given students exam and coursework marks, the program calculates the mark for each module.

Students take six modules in Stage 1: CSC1021 Programming I, CSC1022 Programming II, CSC1023 The Software Engineering Professional which has coursework only, CSC1024 Computer Architecture, CSC1025 Mathematics for Computer Science, CSC1026 Website Design and Construction.

As written in the program specification, each module mark is calculated as follows:

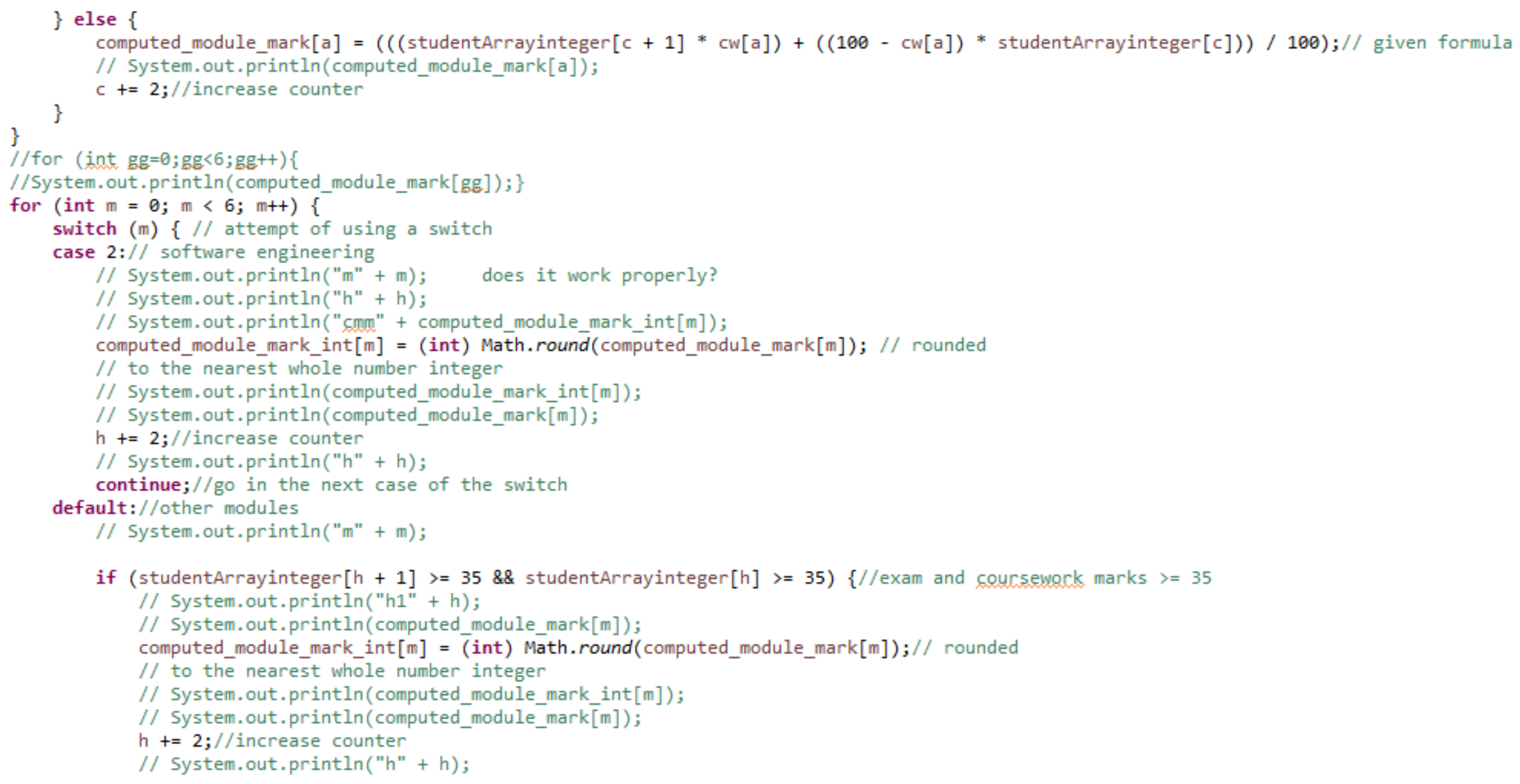
((coursework mark * coursework weighting) + (examination mark * (100 - coursework weighting))) / 100.

As long as I needed to use a considerable number of 'for loops', I tested it to verify there was no error in calculations.



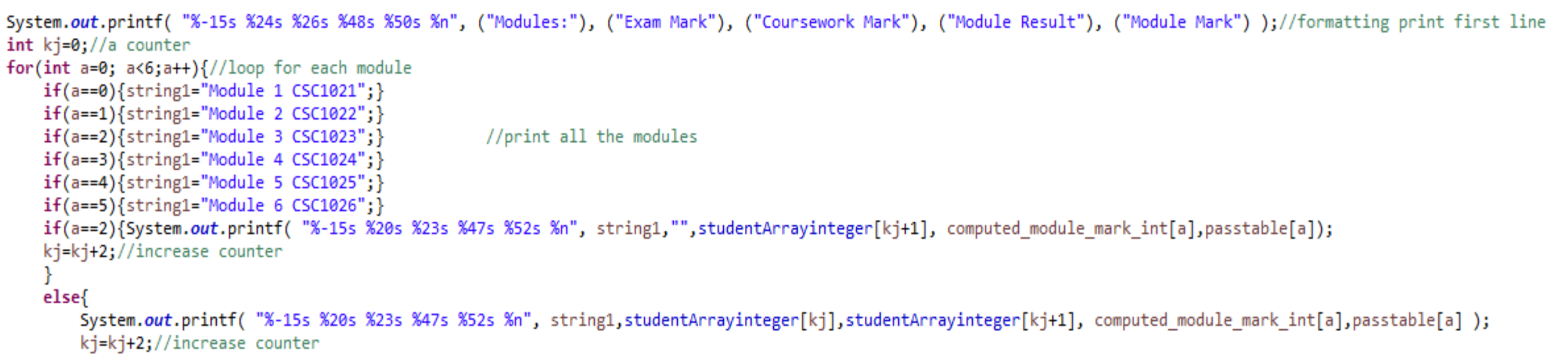
```
public void draw (int[] computed_module_mark_int){ //draws thin bars based on the values in
    /*for (int i=0; i<studentArrayinteger.length; i++)
    {
    //int [] aaa = new int [0];
    //aaa[0]=studentArrayinteger[0];                    //1' attempt to test if it worked
    System.out.println(studentArrayinteger[i]);
    }*/
```

A good way of testing is to put a ‘print statement’ after the code and follow the printed data until you find where the error is if the result is not the one expected.



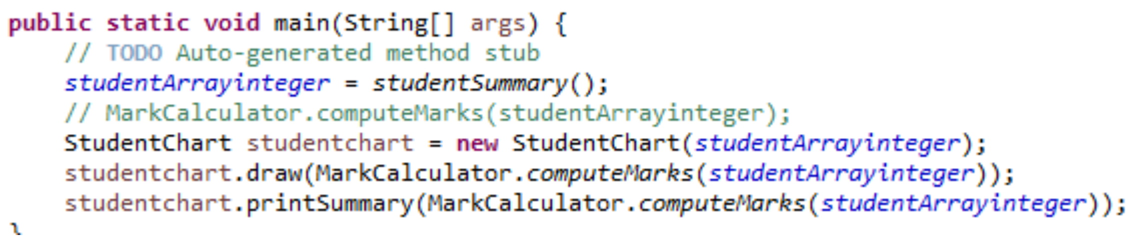
```
    } else {
        computed_module_mark[a] = (((studentArrayinteger[c + 1] * cw[a]) + ((100 - cw[a]) * studentArrayinteger[c])) / 100);// given formula
        // System.out.println(computed_module_mark[a]);
        c += 2;//increase counter
    }
}
//for (int gg=0;gg<6;gg++){
//System.out.println(computed_module_mark[gg]);}
for (int m = 0; m < 6; m++) {
    switch (m) { // attempt of using a switch
    case 2:// software engineering
        // System.out.println("m" + m);     does it work properly?
        // System.out.println("h" + h);
        // System.out.println("cmm" + computed_module_mark_int[m]);
        computed_module_mark_int[m] = (int) Math.round(computed_module_mark[m]); // rounded
        // to the nearest whole number integer
        // System.out.println(computed_module_mark_int[m]);
        // System.out.println(computed_module_mark[m]);
        h += 2;//increase counter
        // System.out.println("h" + h);
        continue;//go in the next case of the switch
    default://other modules
        // System.out.println("m" + m);

        if (studentArrayinteger[h + 1] >= 35 && studentArrayinteger[h] >= 35) {//exam and coursework marks >= 35
            // System.out.println("h1" + h);
            // System.out.println(computed_module_mark[m]);
            computed_module_mark_int[m] = (int) Math.round(computed_module_mark[m]);// rounded
            // to the nearest whole number integer
            // System.out.println(computed_module_mark_int[m]);
            // System.out.println(computed_module_mark[m]);
            h += 2;//increase counter
            // System.out.println("h" + h);
```

After a few attempts to know if the compiled program was correct, the result was a formatted table with all the exact values.



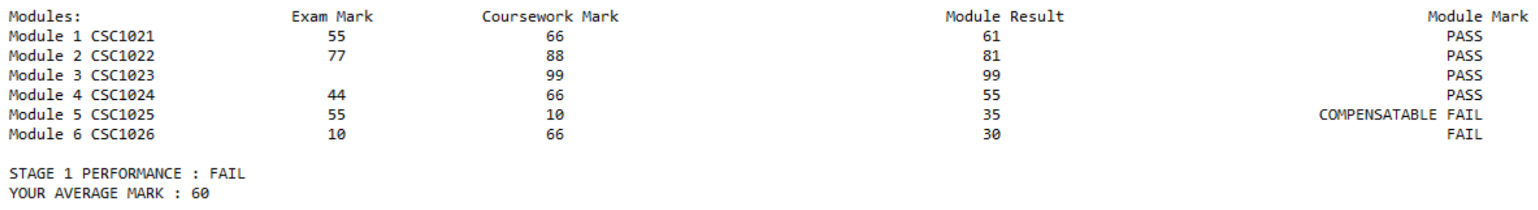
```
System.out.printf( "%-15s %24s %26s %48s %50s %n", ("Modules:"), ("Exam Mark"), ("Coursework Mark"), ("Module Result"), ("Module Mark") );//formatting print first line
int kj=0;//a counter
for(int a=0; a<6;a++){//loop for each module
    if(a==0){string1="Module 1 CSC1021";}
    if(a==1){string1="Module 2 CSC1022";}
    if(a==2){string1="Module 3 CSC1023";}          //print all the modules
    if(a==3){string1="Module 4 CSC1024";}
    if(a==4){string1="Module 5 CSC1025";}
    if(a==5){string1="Module 6 CSC1026";}
    if(a==2){System.out.printf( "%-15s %20s %23s %47s %52s %n", string1,"",studentArrayinteger[kj+1], computed_module_mark_int[a],passtable[a]);
    kj=kj+2;//increase counter
    }
    else{
        System.out.printf( "%-15s %20s %23s %47s %52s %n", string1,studentArrayinteger[kj],studentArrayinteger[kj+1], computed_module_mark_int[a],passtable[a] );
        kj=kj+2;//increase counter
```

The java software asks users to insert for each module exam and coursework mark to use these inputs in other methods.



```java
public static void main(String[] args) {
    // TODO Auto-generated method stub
    studentArrayinteger = studentSummary();
    // MarkCalculator.computeMarks(studentArrayinteger);
    StudentChart studentchart = new StudentChart(studentArrayinteger);
    studentchart.draw(MarkCalculator.computeMarks(studentArrayinteger));
    studentchart.printSummary(MarkCalculator.computeMarks(studentArrayinteger));
}
```

Inserting random values and calculating the module mark with the given formula, we can compare the table with our result.

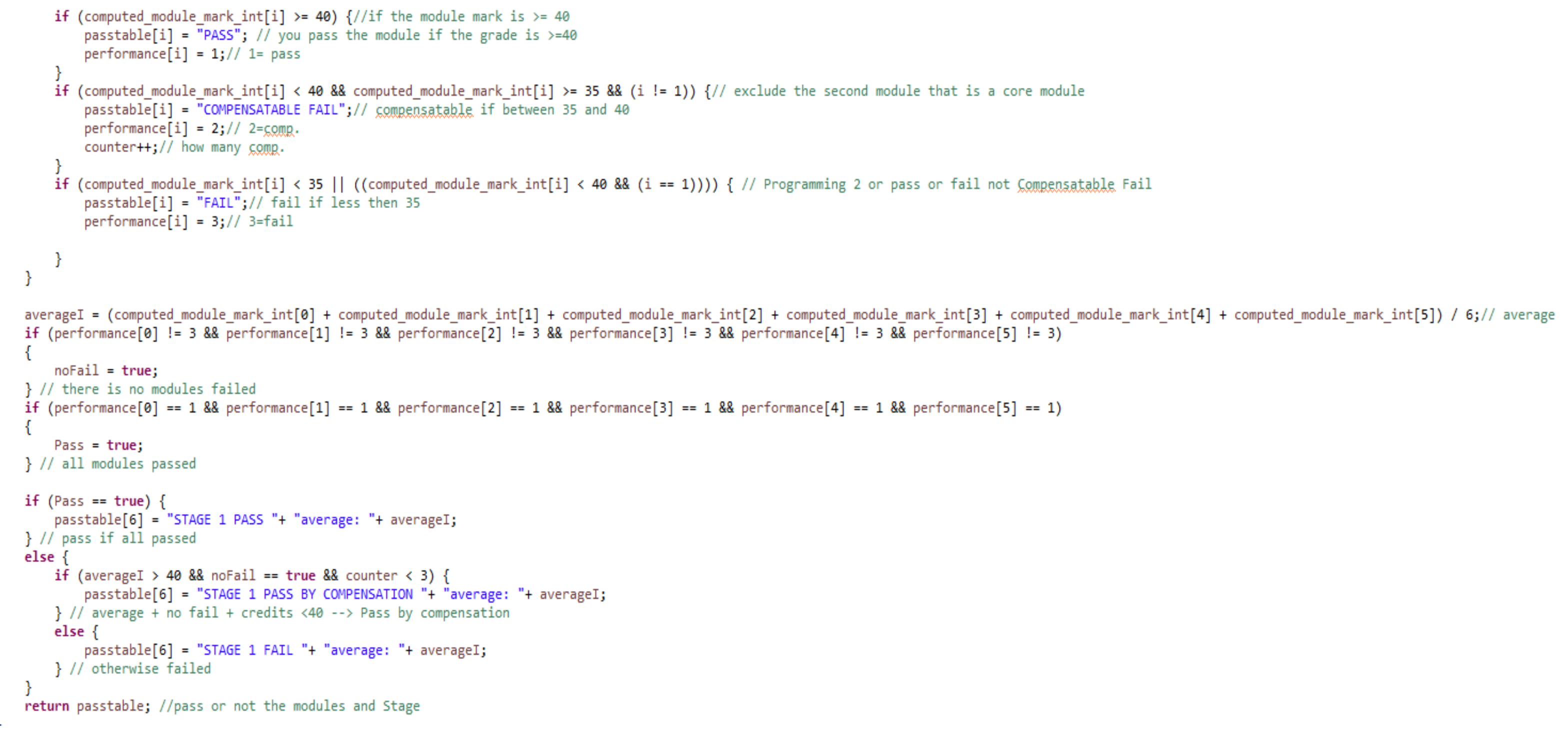


```
Modules:                           Exam Mark             Coursework Mark                                          Module Result                                             Module Mark
Module 1 CSC1021                       55                         66                                                  61                                                      PASS
Module 2 CSC1022                       77                         88                                                  81                                                      PASS
Module 3 CSC1023                                                  99                                                  99                                                      PASS
Module 4 CSC1024                       44                         66                                                  55                                                      PASS
Module 5 CSC1025                       55                         10                                                  35                                        COMPENSATABLE FAIL
Module 6 CSC1026                       10                         66                                                  30                                                      FAIL

STAGE 1 PERFORMANCE : FAIL
YOUR AVERAGE MARK : 60
```

The student passes the module if the module mark is at least 40; the student passes by compensation if the module is non-core, (the only core module is Programming II) and module mark is less than 40 but at least 35 or otherwise the module is computed as fail.

Once all the module marks have been determined, the user can know if Stage one is passed, if and only if all modules are recorded as a pass.

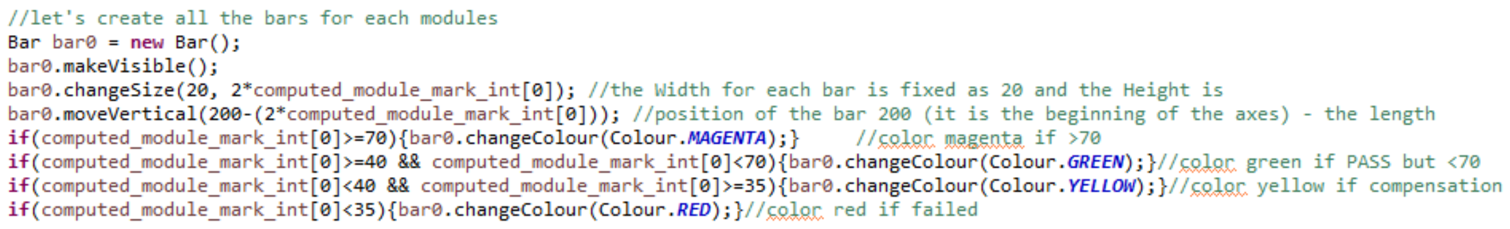
The stage is passed by compensation, if the Stage average is at least 40 and no module is recorded as a Fail, and there are no more than three modules recorded as a compensatable fail or failed otherwise.



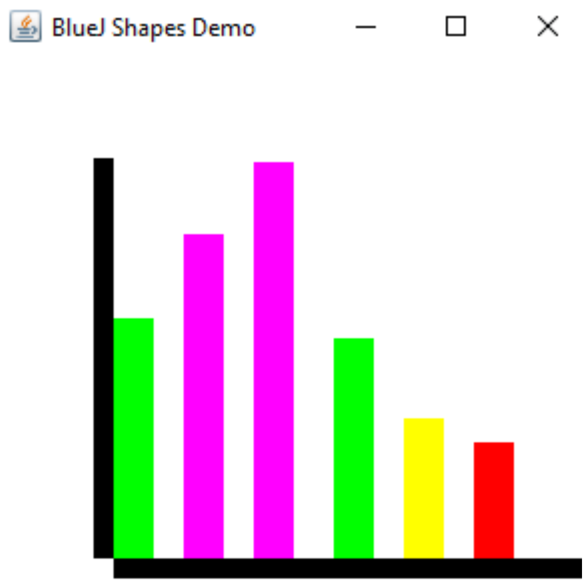
```
    if (computed_module_mark_int[i] >= 40) {//if the module mark is >= 40
        passtable[i] = "PASS"; // you pass the module if the grade is >=40
        performance[i] = 1;// 1= pass
    }
    if (computed_module_mark_int[i] < 40 && computed_module_mark_int[i] >= 35 && (i != 1)) {// exclude the second module that is a core module
        passtable[i] = "COMPENSATABLE FAIL";// compensatable if between 35 and 40
        performance[i] = 2;// 2=comp.
        counter++;// how many comp.
    }
    if (computed_module_mark_int[i] < 35 || ((computed_module_mark_int[i] < 40 && (i == 1)))) { // Programming 2 or pass or fail not Compensatable Fail
        passtable[i] = "FAIL";// fail if less then 35
        performance[i] = 3;// 3=fail

    }
}

averageI = (computed_module_mark_int[0] + computed_module_mark_int[1] + computed_module_mark_int[2] + computed_module_mark_int[3] + computed_module_mark_int[4] + computed_module_mark_int[5]) / 6;// average
if (performance[0] != 3 && performance[1] != 3 && performance[2] != 3 && performance[3] != 3 && performance[4] != 3 && performance[5] != 3)
{
    noFail = true;
} // there is no modules failed
if (performance[0] == 1 && performance[1] == 1 && performance[2] == 1 && performance[3] == 1 && performance[4] == 1 && performance[5] == 1)
{
    Pass = true;
} // all modules passed

if (Pass == true) {
    passtable[6] = "STAGE 1 PASS "+ "average: "+ averageI;
} // pass if all passed
else {
    if (averageI > 40 && noFail == true && counter < 3) {
        passtable[6] = "STAGE 1 PASS BY COMPENSATION "+ "average: "+ averageI;
    } // average + no fail + credits <40 --> Pass by compensation
    else {
        passtable[6] = "STAGE 1 FAIL "+ "average: "+ averageI;
    } // otherwise failed
}
return passtable; //pass or not the modules and Stage
```

The stage result affects the second part of the program in which then, using the Bar class, it draws coloured bars with the height depending on the module mark.

Failing marks should be coloured red, Compensating marks yellow, Pass marks green and First class marks (70 or over) magenta.

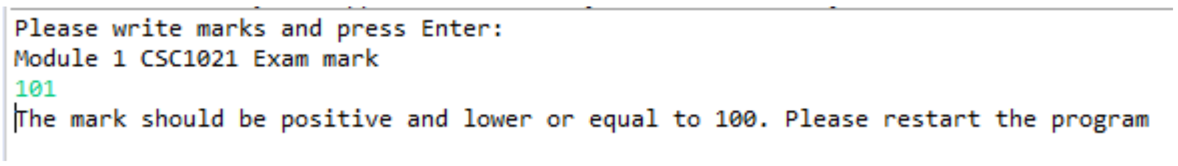


```
//let's create all the bars for each modules
Bar bar0 = new Bar();
bar0.makeVisible();
bar0.changeSize(20, 2*computed_module_mark_int[0]); //the Width for each bar is fixed as 20 and the Height is
bar0.moveVertical(200-(2*computed_module_mark_int[0])); //position of the bar 200 (it is the beginning of the axes) - the length
if(computed_module_mark_int[0]>=70){bar0.changeColour(Colour.MAGENTA);}     //color magenta if >70
if(computed_module_mark_int[0]>=40 && computed_module_mark_int[0]<70){bar0.changeColour(Colour.GREEN);}//color green if PASS but <70
if(computed_module_mark_int[0]<40 && computed_module_mark_int[0]>=35){bar0.changeColour(Colour.YELLOW);}//color yellow if compensation
if(computed_module_mark_int[0]<35){bar0.changeColour(Colour.RED);}//color red if failed
```

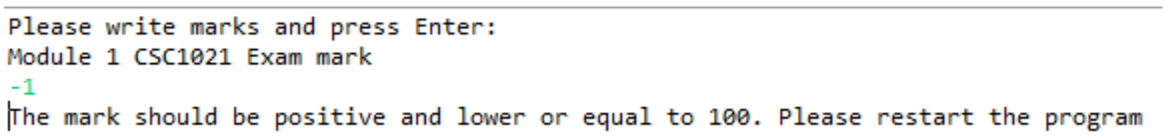


Verify the output doing some maths

First, make sure that the numbers allowed are positive and lower or equal than 100.



```
Please write marks and press Enter:
Module 1 CSC1021 Exam mark
101
The mark should be positive and lower or equal to 100. Please restart the program
```



```
Please write marks and press Enter:
Module 1 CSC1021 Exam mark
-1
The mark should be positive and lower or equal to 100. Please restart the program
```

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
1°	10	30	10	30	10	10	30	10	30	10	30
2°	10	80	10	80	30	10	80	10	80	10	80
3°	80	10	80	10	80	80	10	80	10	80	10
4°	36	36	78	96	48	37	35	99	54	67	84
5°	35	70	35	70	70	35	70	35	70	35	70
6°	90	80	36	40	70	90	80	70	90	80	70

MODULE MARK										
	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026
1°	20		18		10	20		14		17
2°	~~45~~	35	~~38~~	35	30	~~45~~	35	24		34.5
3°	~~45~~	35	~~52~~	35	80	~~45~~	35	~~66~~	35	55.5
4°	36		85.2		48	36		90		72.95
5°	52.5		49		70	52.5		42		47.25
6°	85		37.6		70	85		74		76.50

First, create a table in order to test every situation of fail, pass by compensation and pass.

Then consider different scenarios:

- All the modules are recorded as Fail and the module mark is less than 35 so the computed mark is the minimum between 35 and the computed mark:

You have 10 for the exam and 30 for CW.

The computed mark based on the different weights (50/50 – 60/40 – 100 – 50/50 – 80/20 – 65/35), as we can read in the table above, are:

((10*50)+(30*50))/100=20;

(10*0.6)+(30*0.4)=18;

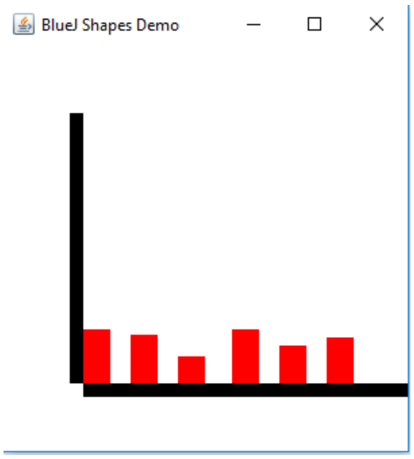
CSC1023 we have coursework only so the mark is the given coursework mark;

(10*0.8)+(30*0.2)=14;

(10*0.65)+(30*0.35)=17.

Since both the marks are less than 35, the rule minimum (35, computed mark) applies.

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
1°	10	30	10	30	10	10	30	10	30	10	30
MODULE MARK											
1°	20		18		10	20		14		17	



```
Modules:                    Exam Mark            Coursework Mark
Module 1 CSC1021                   10                         30
Module 2 CSC1022                   10                         30
Module 3 CSC1023                                              10
Module 4 CSC1024                   10                         30
Module 5 CSC1025                   10                         30
Module 6 CSC1026                   10                         30

STAGE 1 FAIL average: 16
```

```
Module Result                                    Module Mark
           20                                           FAIL
           18                                           FAIL
           10                                           FAIL
           20                                           FAIL
           14                                           FAIL
           17                                           FAIL
```

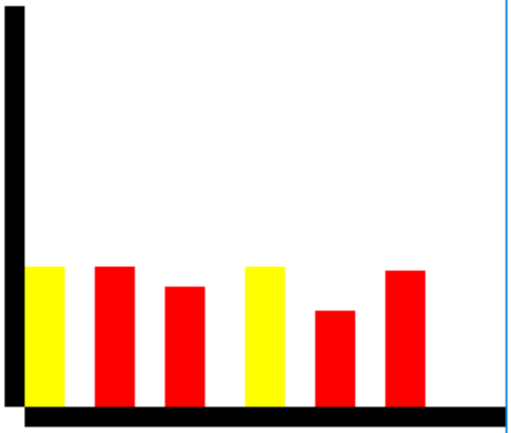
- Two modules are recorded as a compensatable fail but the module mark is greater than 35 so the computed mark is 35 :

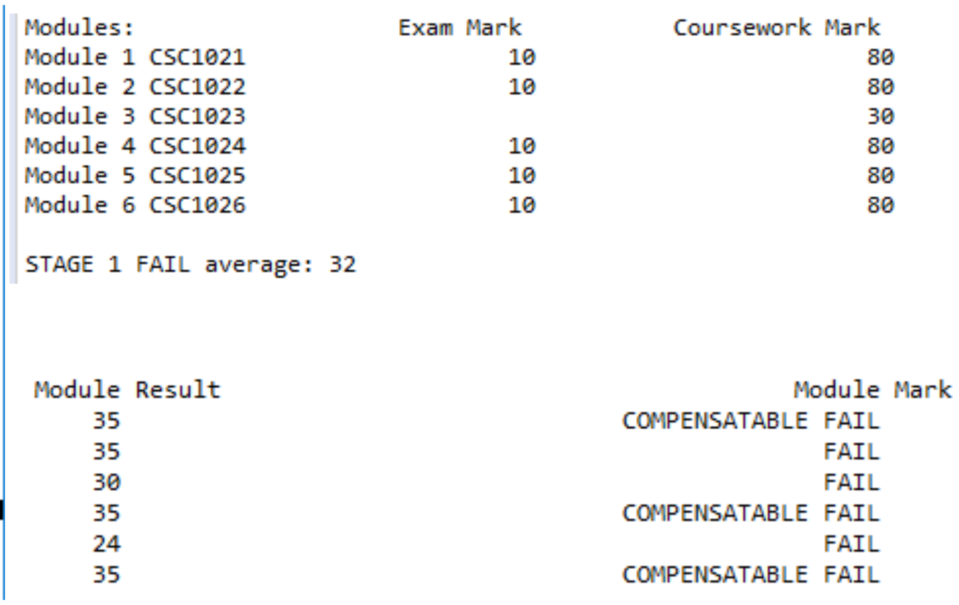
If only one of two marks is less than 35 (Exam mark = 10, Coursework mark=80). The computed mark (based on 50/50) is (10*.5+80*.5)=45.

Since the Exam mark is less than 35, the rule minimum (35, computed mark) applies, therefore your mark would be minimum(35,45)=35. This module would be a Compensatable fail if it is non-core.

In the table, you can read both of the value, the one computed with the formula (marked with double strikethrough) and the one after the filter (minimum).

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
2°	10	80	10	80	30	10	80	10	80	10	80
MODULE MARK											
2°	~~45~~	35	~~38~~	35	30	~~45~~	35	24		34.5	





```
Modules:                    Exam Mark           Coursework Mark
Module 1 CSC1021                   10                        80
Module 2 CSC1022                   10                        80
Module 3 CSC1023                                             30
Module 4 CSC1024                   10                        80
Module 5 CSC1025                   10                        80
Module 6 CSC1026                   10                        80

STAGE 1 FAIL average: 32


 Module Result                                          Module Mark
     35                                       COMPENSATABLE FAIL
     35                                                     FAIL
     30                                                     FAIL
     35                                       COMPENSATABLE FAIL
     24                                                     FAIL
     35                                       COMPENSATABLE FAIL
```

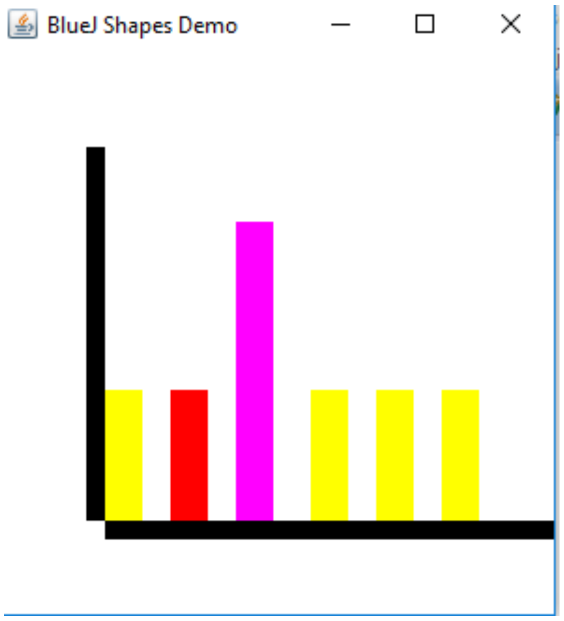
- Compare the second test with this one in order to understand how different weights work. The marks are the same (10 and 80 but now the one which was for the exam is for coursework and vice versa:

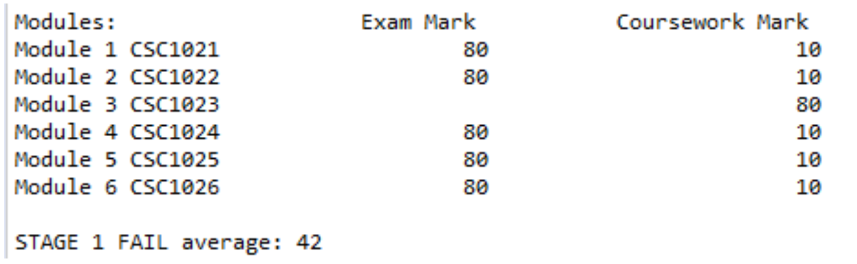
In the third test the results are really similar to the previous marks and for the last module the computed result (highlighted) is a floating-point number and it is stored in a more accurate double array to avoid each possible error (double 64 bits maximum value 10^308 vs integer 32 bits 2*10^9.)

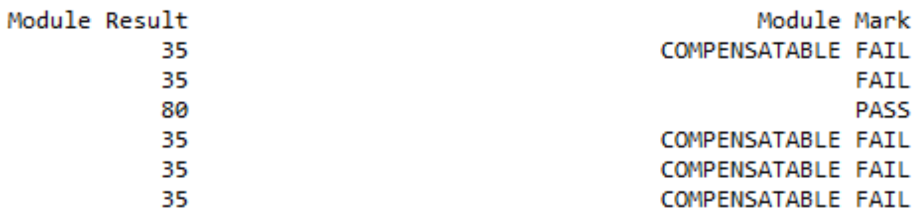
This double then is rounded up into an integer.

The stage result is computed as Fail because there are more than two modules recorded as compensatable fail and the core module, the only module that can only be passed of failed is 35.

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
3°	80	10	80	10	80	80	10	80	10	80	10
MODULE MARK											
3°	~~45~~	35	~~52~~	35	80	~~45~~	35	~~66~~	35	~~55.5~~	35




```
Modules:                     Exam Mark          Coursework Mark
Module 1 CSC1021                    80                       10
Module 2 CSC1022                    80                       10
Module 3 CSC1023                                             80
Module 4 CSC1024                    80                       10
Module 5 CSC1025                    80                       10
Module 6 CSC1026                    80                       10

STAGE 1 FAIL average: 42
```

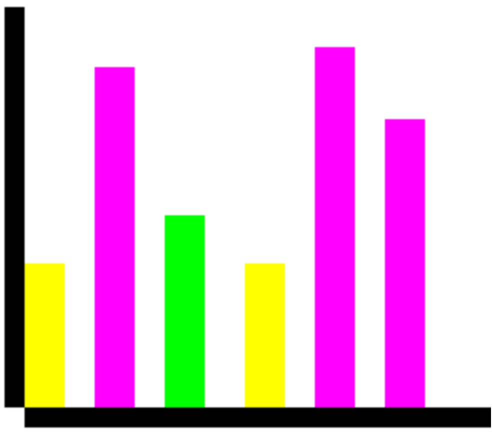

```
Module Result                                       Module Mark
           35                                COMPENSATABLE FAIL
           35                                              FAIL
           80                                              PASS
           35                                COMPENSATABLE FAIL
           35                                COMPENSATABLE FAIL
           35                                COMPENSATABLE FAIL
```

- The Stage is Pass By Compensation because one or two modules are recorded as Compensatable Fail.

If the mark is less than 40 but at least 35, the module (non-core) is recorded as Compensatable Fail.

Then the Stage result can be computed as Pass By Compensation, if the Stage average is at least 40, and there are no more than two modules recorded as a Compensatable Fail.

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
4°	36	36	78	96	48	37	35	99	54	67	84
MODULE MARK											
4°	36		85.2		48	36		90		72.95	



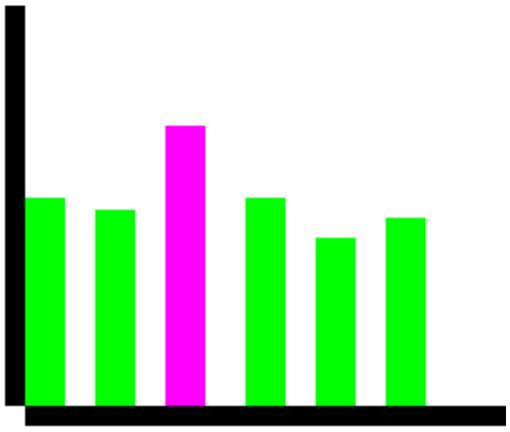
```
Module Result                                        Module Mark
     36                                        COMPENSATABLE FAIL
     85                                                      PASS
     48                                                      PASS
     36                                        COMPENSATABLE FAIL
     90                                                      PASS
     73                                                      PASS
```

```
Modules:                         Exam Mark            Coursework Mark
Module 1 CSC1021                    36                      36
Module 2 CSC1022                    78                      96
Module 3 CSC1023                                            48
Module 4 CSC1024                    37                      35
Module 5 CSC1025                    99                      54
Module 6 CSC1026                    67                      84

STAGE 1 PERFORMANCE : PASS BY COMPENSATION
YOUR AVERAGE MARK : 61
```

- In this example all the modules are recorded as Pass with one greater or equal to 70:

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
5°	35	70	35	70	70	35	70	35	70	35	70
MODULE MARK											
5°	52.5		49		70	52.5		42		47.25	

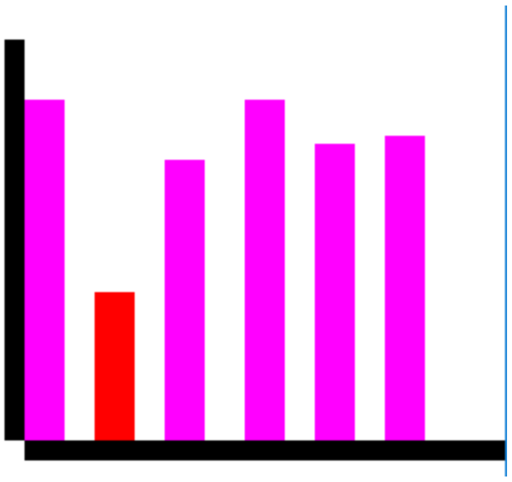
```
Modules:                     Exam Mark          Coursework Mark
Module 1 CSC1021                35                     70
Module 2 CSC1022                35                     70
Module 3 CSC1023                                       70
Module 4 CSC1024                35                     70
Module 5 CSC1025                35                     70
Module 6 CSC1026                35                     70

STAGE 1 PASS average: 52
```



```
Module Result                                   Module Mark
    53                                             PASS
    49                                             PASS
    75                                             PASS
    53                                             PASS
    42                                             PASS
    47                                             PASS
```

- In this case, the modules greater than 70 are recorded as First class marks but Stage is recorded as Fail because CSC1022 cannot be recorded as a Compensatable fail :

	CSC1021		CSC1022		CSC1023	CSC1024		CSC1025		CSC1026	
	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark	Exam Mark	Cours. Mark
6°	90	80	36	40	70	90	80	70	90	80	70
MODULE MARK											
6°	85		37.6		70	85		74		76.5	



```
Modules:                      Exam Mark            Coursework Mark
Module 1 CSC1021                 90                      80
Module 2 CSC1022                 36                      40
Module 3 CSC1023                                         70
Module 4 CSC1024                 90                      80
Module 5 CSC1025                 70                      90
Module 6 CSC1026                 80                      70

STAGE 1 PERFORMANCE : FAIL
YOUR AVERAGE MARK : 71


Module Result                                   Module Mark
    85                                              PASS
    38                                              FAIL
    70                                              PASS
    85                                              PASS
    74                                              PASS
    77                                              PASS
```

Conclusion

Upon providing mathematical tests with the relative calculations and testing all possible outcome and scenarios that might impact the capability of the program, we can with certainty agree that the program is indeed working as intended.

Developing it was not easy because I had to integrate my own knowledge and how much I learned during the lectures with an in-depth research in books and on the internet.

To successfully create this software, I have followed step-by-step all the requirements even though I have often found some difficulties since English is not my first language.

Another problem was working with arrays, creating a single array to store the input of the user, then to process all the marks in the various classes using the 'return statement' correctly for each method because it was often necessary to return more than one variable used simultaneously by multiple methods.

One of the main difficulties I came across was calculating the module mark correctly with the formula and using it to check the requirement to determine the outcome of the Stage.

During all the practical exercises I learned how to do an accurate problem-solving work by looking for errors in saving data while working with arrays using 'for loops' and 'if statements' printing on the console all the values to find out each possible error.

On the other hand, I succeed in using the 'switch statement' and creating a constructor learning the bases of object-oriented development, assigning objects like bars the right attributes (e.g. colour and size.)