Odd/Even/Compact\* Semester (year)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work****)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | |  |  |
| **Student Information**: **Surname** | | | | **Given Names**  **Sefira** | **Student ID Number**  **2001586155** |
| 1. **Karina**  2.  3.  4.  5. | |  |  |
|  |  |
| **Course Code** | **: COMP6335** |  |  | **Course Name** | **: Intro To Programming** |
| **Class** | **: L1AC** |  |  | **Name of Lecturer(s)** | **:** 1. **Ida Bagus Kerthyayana** |
|  |  |  |  |  | 2. **Stavin Deeswe** |
| **Major** | **: CS** |  |  |  |  |
| **Title of Assignment**  (if any) | **: Mid Report Card Simulator** |  |  |  |  |
| **Submission Pattern** |  |  |  |  |  |
| **Due Date** | **: 6-11-16** |  |  | **Submission Date** | **: 6-11-16** |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. 2. 3.

4.

5.

*etc.* \*) Delete the inappropriate option

**MID REPORT CARD SIMULATOR**

**SECTION 1 : DESCRIPTION**

Purpose of Program: To make the process of making report cards easier

Basically, this is a program to calculate student’s test scores into the scores in the mid report card. The steps of the program will go like the following statements bellow :

1. The program will ask the user to enter the amount of students and their names.
2. User will have to input the scores of the students on their first exam.
3. The program will determined if there will be a re test or not. Retest will be held if the average score for the test is bellow the minimum score of 75.
4. If there’s a retest, go back to step 2
5. If there’s no retest , the program will show the name of the students that have to go through remedy and ask the user to input their remedy scores.
6. After step 4 or 5, the user will see all the students final scores for the 1st test. (if retest replace the scores with the new one , the max scores will be 100,if remedy ,replace the scores with the new one , the max scores will be the minimum passing score .
7. After done with the first exam ,the program will asked for the 2nd exam score and process it the same way with the 1st exam.
8. Then , each student’s average score for both tests will be calculated
9. After that , the program will ask user to input the mid test’s scores . But , for mid test , there’s no remedy ,only retest
10. The program will calculate the final scores for the report , that being 70% of the average test score and 30% of the mid term .
11. The user will see the report card scores of each students.

**SECTION 2.A : DESIGN / PLAN**

main

Menu

1

3

2

exit

Input students

rules

Input names

Input scores 1

Input scores 2

Calculate average scores

Input midscores

Retest/ not

Remedy /not

Retest/ not

Remedy/ not

Retest/ not

Calculate final mid scores

Report card scores

The program have a class named Data in the header file . The private consists of the amount of students(int) and array of their names(string) while the public consist of :

* Data() 🡪 constructor
* void setStudents
* void setName
* int getStudents
* string getName
* void rules
* void calcScores
* void calcScores2
* void countAv
* void midTerm
* void reportScore

When the program is being run, the first thing that user see is the menu that has 3 choice . The first choice is to see scoring rules, the second choice is to start scoring and the last choice is to exit the program. After choosing one of them , because of the do-while loop , the menu will reappeared later once the choice is done.Each choice of menu has a different functions assign to it from class Data. The first choice will show the content of void rules() .

The second choice (start scoring) is the main purpose of this program. When user choose it , the first thing the program do is asking for the amount of student and that amount of student input will be used for almost all for loops in this program. After that , the program will ask user to input the names of the student with the amount of student input earlier , then the names will be stored in arrays .

Next is asking for the test scores and count the average . with if statement , the program will decide whether the retest or remedy test will be held or not , if yes , it will ask for the new score , if not then it will continue to the second and mid test scores. And it will do the same for the second test and mid test (but there’s no remedy for mid test). After all the calculation of all the tests , all students’s final score for the mid report will be shown .

**SECTION 2B : EXPLANATION OF EACH FUNCTION/CLASS**

**Data()**  : Constructor

**void setStudents** : mutator to set the amount of students

**void setName** : mutator to set student’s name

**int getStudents** : accessor to return the value of student

**string getName** : accessor to return the value of student’s names

**void rules**

This function will appear when user choose the 1st choice in the menu. The content is

the rule of scoring in this program.

**void calcScores**

This function ask user to input student’s 1st test score, determine

wheter there’s retest/remedy/not and calculate the final score for the test. Inside

this function , there’s a for loop that is used to get the score , do while loop as the

input validation , 2 if statements to determined retest/remedy and a for loop to

show the final scores for this test.

**void calcScores2**

To get the 2nd test score, works exacly the same as void calcScores.

**void countAv**

To count the average score of each student with for loop and the calculation :

mean[i]=(score[i]+score2[i])/2 ;

**void midterm**

A function to ask and calculate the mid test score. Works exacly like calcScore and calcScore2 , but without remedy test .

**void reportScore**

Last function , to count the final score of the mid report card with the calculation:

finalMidScore [i]= 0.3\*mean[i]+0.7\*midScore[i] ;

After that the all the scores will be shown starting from the average test scores , mid test scores and the final mid report scores .

**SECTION 3A : EVERYTHING LEARNED DURING THE PROCESS**

This project started early days of this semester when the knowledged of programming and c++ were very limited. But through making this program, many new things were learned from a simple thing like using loop to some more advance things like using class. Things learned from this project includes :

* **make arrays and for loop worked together**

to assign something inside the loop (ex: name) there’s other way to do it other than do it manually like name[0]=”who” ,name[1]=”who2” , the shorter way to do that is by using for loop .ex :

for(int i=0 ; i<5 ; i++)

{

cout<<”name” << i<<” = “ ; cin>>name[i] ;

}

* **How to use arrays as function’s argument**

To declare array in function’s argument , a [] is needed after the data type .

Ex : void something(int []);

* **Using while and do-while loops to make input validation**

Input validation is something that appeared every time a condition given is fullfiled . To make this, loop is needed so that the error massage or others will appear over and over again as long as the condition given is fullfiled.

Ex :

while(retestScore2[i]>100 || retestScore2[i]<0)

{

cout << "input "<<name[i]<<"'s retest's score (0-100) = ";

cin>>retestScore2[i];

}

* **The usage of accessor and mutator in class**

A mutator is a function that allows for editing of the contents of the private member .  it have the value to be assigned to the data member.

Ex :

void setStudents(int humans)

{

students=humans ;

}

An accessor is a function that allows modifications of the data members in class. It must return a value and has no parameter.

Ex:

Int getStudents()

{

return students ;

}

**SECTION 3B : THE PROBLEMS THAT WERE OVERCOME**

On the second week , some difficulties was found in determined student’s score that was bellow the minimum score and asking the user for their new scores. That time , some methods were being tried , including :

for (int i=0 ; i<students ; i++)

{

if (score[i]<minScore)

{

name[i]=noPassName[i];

score[i]=noPassScore[i];

}

cout <<noPassName[i]<<endl;

}

And that’s definitely wrong . But this problem was solved in week 3 by using :

if (score[i]< minScore)

{

cout << "Please input " << name[i] << "'s remedy score =";

cin >>remedyScore[i];

if(remedyScore[i]<score[i])

remedyScore[i]=score[i];

else if(remedyScore[i]>score[i])

{

if(remedyScore[i]<minScore)

score[i]=remedyScore[i];

else if (remedyScore[i]>minScore)

remedyScore[i] = minScore ;

score[i]=remedyScore[i];

}

remedyScore[i]=score[i];

}

On the 4rd week , the way to calculate student’s average test scores can’t be found , turns out it was because the array that was used to store the 2nd test scores the same with the arrays that was used to store the 1st test score (score[i]) ,so when the user input the 2nd test scores, the 1st test score that was already inside the arrays were being replaced by the new scores . This problem was solved by making separated arrays for the 1st and 2nd test scores (score[i] and score2[i]) ,then count the average with : mean[i]=(score[i]+score2[i])/2;

Another Problem on the final week was how to make the menu reappeared after the user choosed something from the menu. The first thing that was being tried was by using the if statement :

if(choice>0 || choice<4)

{

cout <<"What do you want to do?"<<endl<<endl;

cout<<"1.View scoring rules\n"<<"2.Start scoring\n" <<"3.Exit"<<endl<<endl;

cout<<"Input your choice (1-3) : " ; cin>>choice ; cout<<endl;

}

But turned out , the right way to do it was by using do-while loop .

Do

{

…...... ;

}

while(choice>0 || choice<4);

**SECTION 4 : Coding**

**finalproject.hpp**

#ifndef finalproject\_hpp

#define finalproject\_hpp

#include<iostream>

#include <string>

using namespace std;

class Data

{

int students;

string name[20];

public :

Data();

void setStudents(int humans);

void setName(string);

int getStudents();

string getName();

void rules();

void calcScores(int,string name[],float score[],float total,float average,float retestScore[],float remedyScore[]);

void calcScores2(int,string name[],float score2[],float total2,float average2,float retestScore2[],float remedyScore2[]);

void countAv(float score[],float score2[],float mean[],string name[]);

void midTerm(int,string name[],float score3[],float total3,float average3,float retestScore3[],float midScore[]);

void reportScore(int,float finalMidScore[],float mean[],float midScore[],string name[]);

};

Data::Data() //constructor

{

students=students ;

}

/////////////////////////////////////////////////////////////////

void Data::setStudents(int humans)

//mutator to set the amount of students into the user input

{

students=humans ;

}

////////////////////////////////////////////////////////////////////

void Data::rules() //the rules of scoring

{

cout <<"RULES OF SCORING FOR MID REPORT : \n"

<<"-------------------------------------\n\n"

<<"1.There will be 2 tests and 1 mid test\n\n"

<<"2.The minimum score is 75\n\n"

<<"3.If the average score of the test is bellow that,retest\n"

<<" will be needed,the max retest score is 100\n\n"

<<"4.If a student get red score but no retest,the student\n"

<<" will undergo a remedy,with the max score of 75\n"

<<" if the remedy score is above 75,the score will\n"

<<" be count as 75,if the remedy score is bellow 75,\n"

<<" the highest score will be taken (ex:score=70 ,remedy\n"

<<" score =60, then 70 will be taken)\n\n"

<<"5.There will be no remedy for the mid test\n\n"

<<"6.The final score for the mid report card is 30% of the\n"

<<" average test score + 70% of the mid score\n"

<<"------------------------------------------------------------\n\n";

}

void Data::setName(string)

//mutator to set student's name

{

string namee[20];

for (int i=0 ; i<students ; i++)

{

name[i]=namee[i];

}

}

///////////////////////////////////////////////////////////////////////

int Data::getStudents()

//accessor to return the value of student

{

return students ;

}

//////////////////////////////////////////////////////////////////////

string Data::getName()

//accessor to return the value of student's names

{

return name[20];

}

///////////////////////////////////////////////////////////////////////

void Data::calcScores(int,string name[],float score[],float total,float average,float retestScore[],float remedyScore[])

/\*the function void calcScores2 and void midTerm

have the exact same explanation as void calcScore ,

the only difference is void midTerm doesn't have remedy\*/

{

const int minScore=75;

for (int i=0 ; i<students ; i++)

// to know student's tests scores

{

cout << "input "<<name[i]<<"'s test score " << " = ";

cin>>score[i];

while(score[i]>100 || score[i]<0)

/\*input validation , if the score is bellow 0 or above 100,

the program will ask the user to reinput\*/

{

cout << "input "<<name[i]<<"'s test score (0-100) " << " = ";

cin>>score[i];

}

total = total + score[i] ;

// count the total score of all students

//(used to count average score later)

}

average = total / students;

// count the average score for the test

cout<<endl<<"average score = "<< average<<endl<<endl;

if(average < minScore)

//if the average above is bellow 75,a retest will be held

{

cout<<"The average score is bellow the minimum score (75)"<<endl

<<"please input students's retest scores"<<endl<<endl;

for (int i=0 ; i<students ; i++)

{

do

/\*input validation , if the score is bellow 0 or above 100,

the program will ask the user to reinput\*/

{

cout << "input "<<name[i]<<"'s retest's score (0-100) = ";

cin>>retestScore[i];

}

while(retestScore[i]>100 || retestScore[i]<0);

if(retestScore[i]<score[i])

/\*the highest score between the retest score

or the previous test score will be used\*/

retestScore[i]=score[i];

else if(retestScore[i]>score[i])

score[i]=retestScore[i];

retestScore[i]=score[i];

}

cout<<endl<<endl;

}

else if (average > minScore)

{

for (int i=0 ; i<students ; i++)

{

if (score[i]< minScore)

{

do

/\*input validation , if the score is bellow 0 or above 100,the program will ask the user to reinput\*/

{

cout << "input "<<name[i]<<"'s retest's score (0-100) = ";

cin>>remedyScore[i];

}

while(remedyScore[i]>100 || remedyScore[i]<0);

if(remedyScore[i]<score[i])

remedyScore[i]=score[i];

/\*if the remedy score is less than previous score,

the highest score will be taken\*/

else if(remedyScore[i]>score[i])

{

if(remedyScore[i]<minScore)

score[i]=remedyScore[i];

/\*if the remedy score is above 75,

the score will be count as 75\*/

else if (remedyScore[i]>minScore)

remedyScore[i] = minScore ;

/\*if the remedy score is above 75,

it will count as 75\*/

score[i]=remedyScore[i];

}

remedyScore[i]=score[i];

}

}

cout<<endl;

}

cout<<"final score for this test : "<<endl;

for (int i=0 ; i<students ; i++)

//to show all scores after remedy/retest

{

cout << name [i] << " : "<<score[i]<<endl;

}

cout<<endl;

}

//=============================================================================//

void Data::calcScores2(int,string name[],float score2[],float total2,float average2,float retestScore2[],float remedyScore2[])

{

const int minScore=75;

for (int i=0 ; i<students ; i++) // to know student's tests scores

{

cout << "input "<<name[i]<<"'s test score " << " = ";

cin>>score2[i];

while(score2[i]>100 || score2[i]<0)

{

cout << "input "<<name[i]<<"'s test score (0-100) " << " = ";

cin>>score2[i];

}

total2 = total2 + score2[i] ;

}

average2 = total2 / students;

cout<<endl<<"average score = "<< average2<<endl<<endl;

if(average2 < minScore)

{

cout<<"The average score is bellow the minimum score (75)"<<endl

<<"please input students's retest scores"<<endl<<endl;

for (int i=0 ; i<students ; i++)

{

do

{

cout << "input "<<name[i]<<"'s retest's score (0-100) = ";

cin>>retestScore2[i];

}

while(retestScore2[i]>100 || retestScore2[i]<0);

if(retestScore2[i]<score2[i])

retestScore2[i]=score2[i];

else if(retestScore2[i]>score2[i])

score2[i]=retestScore2[i];

retestScore2[i]=score2[i];

}

cout<<endl<<endl;

}

else if (average2 > minScore)

{

for (int i=0 ; i<students ; i++)

{

if (score2[i]< minScore)

{

do

{

cout << "Please input " << name[i] << "'s remedy score =";

cin >>remedyScore2[i];

}

while(remedyScore2[i]>100 || remedyScore2[i]<0);

if(remedyScore2[i]<score2[i])

remedyScore2[i]=score2[i];

else if(remedyScore2[i]>score2[i])

{

if(remedyScore2[i]<minScore)

score2[i]=remedyScore2[i];

else if (remedyScore2[i]>minScore)

remedyScore2[i] = minScore ;

score2[i]=remedyScore2[i];

}

remedyScore2[i]=score2[i];

}

}

cout<<endl;

}

cout<<"final score for this test : "<<endl;

for (int i=0 ; i<students ; i++)

{

cout << name [i] << " : "<<score2[i]<<endl;

}

cout<<endl;

}

//=========================================================================//

void Data::countAv(float score[],float score2[],float mean[],string name[])

{

for (int i=0 ; i<students ; i++)

{

mean[i]=(score[i]+score2[i])/2;

//to calculate the average of the 1st and 2nd test

cout<<name[i] <<"'s average test score = " << mean[i]<<endl;

}

}

//==========================================================================//

void Data::midTerm(int,string name[],float score3[],float total3,float average3,float retestScore3[],float midScore[])

{

const int minScore=75;

for (int i=0 ; i<students ; i++) // to know student's tests scores

{

cout << "input "<<name[i]<<"'s mid test score (0-100) " << " = ";

cin>>midScore[i];

while(midScore[i]>100 || midScore[i]<0)

{

cout << "input "<<name[i]<<"'s test score (0-100) " << " = ";

cin>>midScore[i];

}

total3 = total3 + midScore[i] ;

}

cout<<endl;

average3 = total3 / students;

cout<<"average score = "<< average3<<endl<<endl;

if(average3 < minScore)

{

cout<<"The average score is bellow the minimum score (75)"<<endl

<<"please input students's retest scores"<<endl<<endl;

for (int i=0 ; i<students ; i++)

{

do

{

cout << "input "<<name[i]<<"'s retest's score (0-100) = ";

cin>>retestScore3[i];

}

while(midScore[i]>100 || midScore[i]<0);

if(retestScore3[i]<midScore[i])

retestScore3[i]=midScore[i];

else if(retestScore3[i]>midScore[i])

midScore[i]=retestScore3[i];

retestScore3[i]=midScore[i];

}

cout<<endl<<endl;

}

}

//============================================================================//

void Data :: reportScore(int,float finalMidScore[],float mean[],float midScore[],string name[])

//to show all the scores that must be inserted to the report card

{

cout<<endl;

cout<<"name\t"<<"Average Score\t"<<"Mid Score\t"<<"Score For Now"<<endl<<endl;

for (int i=0 ; i<students ; i++)

{

finalMidScore [i]= 0.3\*mean[i]+0.7\*midScore[i] ;

//to calculate final score for the report

cout<<name[i]<<"\t"<<mean[i]<<"\t\t"<<midScore[i]<<"\t\t"<<finalMidScore[i]<<endl;

}

}

#endif /\*finalproject\_hpp\*/

**main.cpp**

// main.cpp

// Final Project

// Sefira Karina-2001586155-Batch 2020

// Copyright © 2016 Sefira Karina. All rights reserved.

#include<iostream>

#include <string>

#include "finalproject.hpp"

using namespace std;

int main()

{

int humans,choice ;

int num=20;

string namee[num] ;

float score[num],average,remedyScore[num],retestScore[num],total;

float score2[num],average2,remedyScore2[num],retestScore2[num],total2,mean[num];

float score3[num],average3,retestScore3[num],total3,midScore[num],finalMidScore[num];

cout<< "WELCOME TO THE MID REPORT CARD SIMULATOR\n"

<<"-----------------------------------------"<<endl<<endl ;

do

//the menu will reappeared after each choice (except 3)

{

cout <<"What do you want to do?"<<endl<<endl;

cout<<"1.View scoring rules\n"<<"2.Start scoring\n"<<"3.Exit"<<endl<<endl;

cout<<"Input your choice (1-3) : " ; cin>>choice ; cout<<endl;

Data input;

if (choice==1)

{

input.rules();

//call fucntion to show the rules of scoring)

}

else if(choice==2)

{

cout << "Enter number of students (max 20) = ";

//input the number of students

cin >> humans ;

while(humans > num)

/\*input validation,if the student input is above 20,

the program will ask for reinput\*/

{

cout << "Enter number of students (max 20) = ";

cin >> humans ;

}

cout<<endl;

input.setStudents(humans);

//call mutator to set the amount of students

for (int i=0 ; i<input.getStudents() ; i++)

// to know students's names

{

cout << "Enter student "<<i+1<<"'s name = " ;

cin >> namee[i] ;

input.setName(namee[i]);

}

cout<<endl ;

cout <<"1st test"<<endl ;

cout<<"------------"<<endl ;

//call the function to input & calculate 1st test's scores

input.calcScores(input.getStudents(),namee,score,total,average,retestScore,remedyScore);

cout<<"====================================================================="<<endl;

cout <<"2nd test"<<endl ;

cout<<"------------"<<endl ;

//call function to input & calculate 2nd test's scores

input.calcScores2(input.getStudents(),namee,score2,total2,average2,retestScore2,remedyScore2);

cout<<"======================================================================"<<endl;

//call function to calculate the average test scores

input.countAv(score,score2,mean,namee);

cout<<"======================================================================"<<endl;

//call function to input & calculate mid test's scores

input.midTerm(input.getStudents(),namee,score3,total3,average3,retestScore3,midScore);

//call function to show the final scores for the mid repord card

input.reportScore(input.getStudents(),finalMidScore,mean,midScore, namee);

}

else if(choice==3)

//to exit the program,user must choose 3 in the menu

{

cout<<"OKAY,BYE!!" ;

return 0;

}

else

cout<<"PLEASE CHOSE FROM 1-3 ONLY\n\n";

}

while(choice>0 || choice<4);

return 0;

}

~~~~END~~~