NAME : SMITA LEKHAK

Write a program that manages a simple student grade calculator with the following requirements. Create a Student class that has:

1. Student name (string)
2. Three subject marks (integers)
3. A basic member function to calculate average

The program should:

1. Accept student details (name and marks) from user input
2. Calculate and display:
   1. Total marks
   2. Average marks
   3. Grade (A for ≥90%, B for ≥80%, C for ≥70%, D for ≥60%, F for <60%)
3. Display a message if any mark is below 0 or above 100

SOLUTION:

#include <iostream>

#include <string>

using namespace std;

class student

{

private:

float mark1,mark2,mark3,total,avg,percent;

string name;

char grade;

public:

void takeinput()

{

cout<<"Enter your full name"<<endl;

getline(cin,name);

cout<<"Enter marks of Subject 1 out of 100"<<endl;

cin>>mark1;

cout<<"Enter marks of Subject 2 out of 100"<<endl;

cin>>mark2;

cout<<"Enter marks of Subject 3 out of 100"<<endl;

cin>>mark3;

}

void calculate()

{

total = mark1 + mark2 + mark3;

avg = total/3;

percent = (total / 300 )\*100;

if (percent>=90)

{

grade = 'A';

}

else if (percent>=80)

{

grade = 'B';

}

else if (percent>=70)

{

grade = 'C';

}

else if (percent>=60)

{

grade = 'D';

}

else

{

grade = 'F';

}

}

void display()

{

cout<<"NAME OF STUDENT: "<<name<<endl;

cout<<"TOTAL MARKS: "<<total<<endl;

cout<<"AVERAGE MARKS: "<<avg<<endl;

cout<<"GRADE: "<<grade<<endl;

}

void check()

{

if (0>mark1 || mark1>100 || 0>mark2 || mark2>100 || 0>mark3 || mark3>100)

{

cout<<"INVALID MARKS";

}

else {

calculate();

display();

}

}

};

int main()

{

student s1;

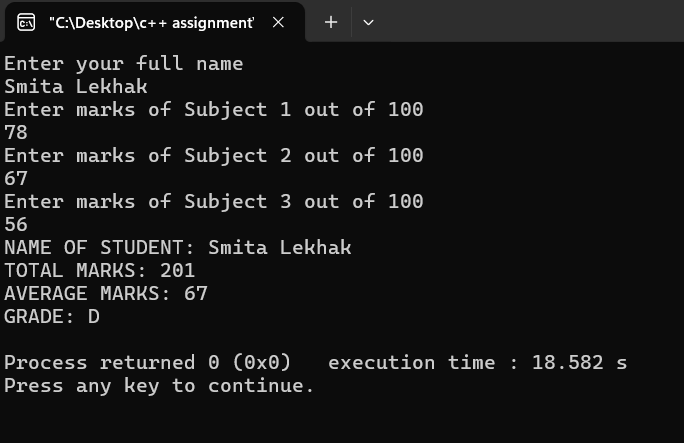
s1.takeinput();

s1.check();

return 0;

}

OUTPUT:



**Task 2: Programming assignments: All questions are mandatory**

1. Write a program with a class Circle having:
   1. Private member: radius (float)
   2. A constructor to initialize radius
   3. A friend function compareTwoCircles that takes two Circle objects and prints which circle has the larger area

SOLUTION:

#include <iostream>

using namespace std;

class Circle

{

private:

float radius1 , radius2 ;

const float pi = 3.14;

float area1, area2;

public:

Circle()

{

radius1 = 0;

radius2 = 0;

}

void input()

{

cout<<"ENTER RADIUS OF CIRCLE 1"<<endl;

cin>>radius1;

cout<<"ENTER RADIUS OF CIRCLE 2"<<endl;

cin>>radius2;

}

void calculate()

{

area1= pi \* radius1 \* radius1;

area2= pi \* radius2 \* radius2;

}

void display()

{

if (radius1>radius2)

{

cout<<"AREA OF CIRCLE 1 : "<<area1 <<endl;

cout<<"AREA OF CIRCLE 2 : "<<area2 <<endl;

cout<<"AREA OF CIRCLE 1 IS GREATER";

}

else {

cout<<"AREA OF CIRCLE 1 : "<<area1 <<endl;

cout<<"AREA OF CIRCLE 2 : "<<area2 <<endl;

cout<<"AREA OF CIRCLE 2 IS GREATER";

}

}

};

int main()

{

Circle c1;

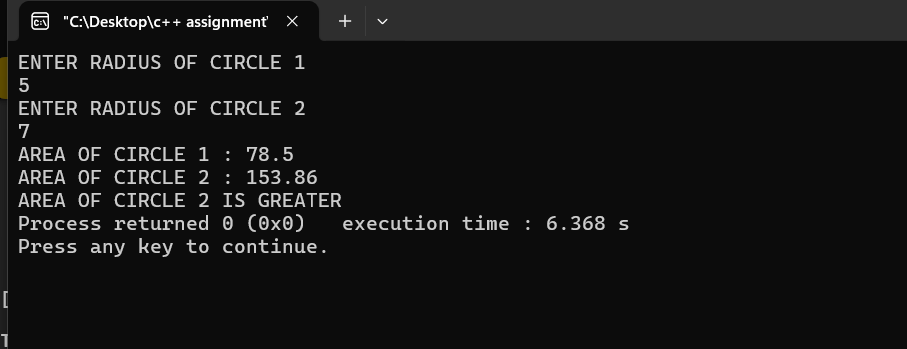
c1.input();

c1.calculate();

c1.display();

}

OUTPUT:

****

1. Create a program with these overloaded functions named findMax:
   1. One that finds maximum between two integers
   2. One that finds maximum between two floating-point numbers
   3. One that finds maximum among three integers
   4. One that finds maximum between an integer and a float **[30 marks]**

**SOLUTION:**

**#include <iostream>**

**using namespace std;**

**class overload**

**{**

**public:**

**void findMax(int a,int b)**

**{**

**if (a>b)**

**{**

**cout<<a<<" > "<< b <<endl;**

**}**

**else**

**{**

**cout<<b<<" > "<< a <<endl;**

**}**

**}**

**void findMax(float a, float b)**

**{**

**if (a>b)**

**{**

**cout<<a<<" > "<< b <<endl;**

**}**

**else**

**{**

**cout<<b<<" > "<< a <<endl;**

**}**

**}**

**void findMax(int a,int b,int c)**

**{**

**if (a>b && a>c)**

**{**

**cout<<a<<" > "<< b <<" and "<< c <<endl;**

**}**

**else if (b>a && b>c)**

**{**

**cout<<b<<" > "<< a<<" and "<< c <<endl;**

**}**

**else**

**{**

**cout<<c<<" > "<< b<<" and "<< a <<endl;**

**}**

**}**

**void findMax(int a , float b)**

**{**

**cout<<a<<" > "<<b<<endl;**

**cout<<b<<" > "<<a<<endl;**

**}**

**};**

**int main()**

**{**

**overload o1;**

**o1.findMax(5,6);**

**o1.findMax(2.2f,1.9f);**

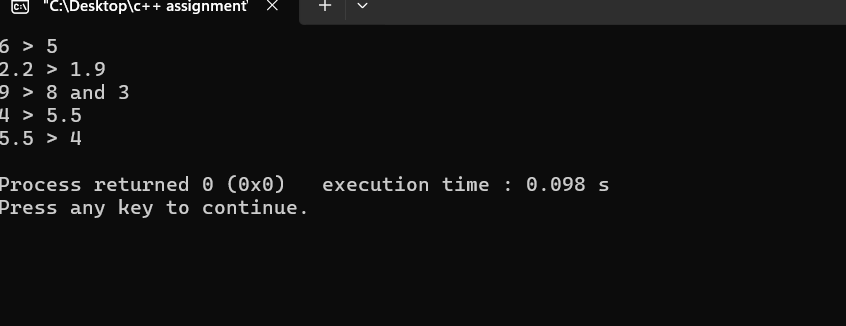
**o1.findMax(8,9,3);**

**o1.findMax(4,5.5f);**

**return 0;**

**}**

**OUTPUT:**

****

**Task 3: Basics of File Handling**

Write a program that reads the titles of 10 books (use an array of 150 characters) and writes them in a binary file selected by the user. The program should read a title and display a message to indicate if it is contained in the file or not.

**SOLUTION:**

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**#include <limits>**

**using namespace std;**

**int main()**

**{**

**char titles[10][150];**

**//open file and check**

**fstream file;**

**file.open("titles.txt", ios::binary | ios::out | ios::trunc );**

**if (!file)**

**{**

**cout<<"ERROR! FILE NOT OPEN"<<endl;**

**return 1;**

**}**

**//enter 10 titles and store**

**for (int i=0; i<10 ; i++)**

**{**

**cout<<"enter title of book "<<endl;**

**cin.getline(titles[i], 150);**

**file.write(titles[i], 150);**

**}**

**cin.ignore(numeric\_limits<streamsize>::max(), '\n');**

**string useri;**

**cout << "ENTER THE BOOK TITLE TO SEARCH: ";**

**getline(cin, useri);**

**// to check if title is there**

**ifstream checkfile("titles.txt",ios::binary);**

**if (!checkfile)**

**{**

**cout<<"Error file not open"<<endl;**

**return 1;**

**}**

**char temp[150];**

**bool found = false;**

**for (int i = 0; i < 10; i++)**

**{**

**checkfile.read(temp, 150);**

**temp[149] = '\0';**

**string current(temp);**

**current = current.substr(0, current.find('\0'));**

**current.erase(current.find\_last\_not\_of(" \t\n\r") + 1);**

**if (current == useri)**

**{**

**cout << "FOUND" << endl;**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout<<"NOT FOUND";**

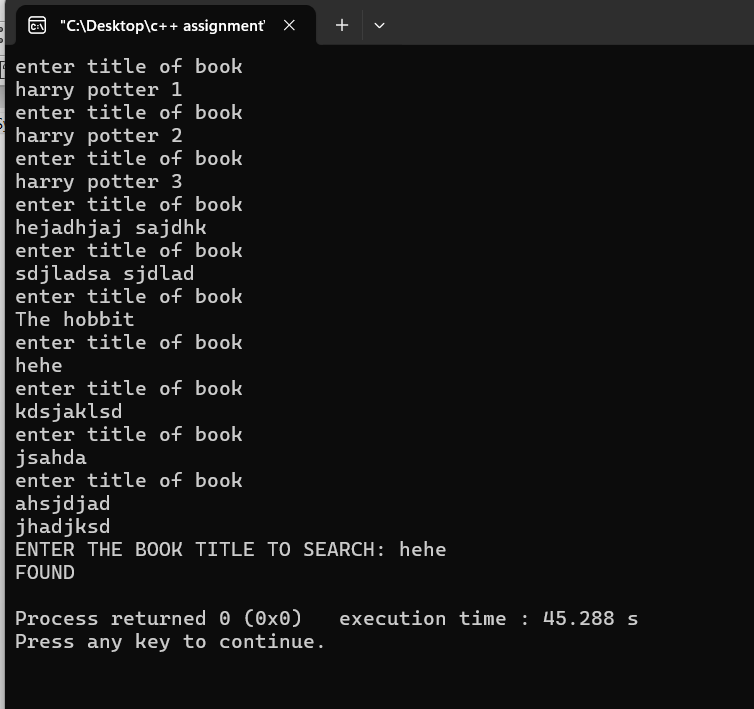
**}**

**checkfile.close();**

**return 0;**

**}**

**OUTPUT:**

****

Create a program that:

1. Reads student records (roll, name, marks) from a text file
2. Throws an exception if marks are not between 0 and 100
3. Allows adding new records with proper validation
4. Saves modified records back to file

**SOLUTION:**

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**using namespace std;**

**class student**

**{**

**private:**

**float mark1,mark2,mark3,roll;**

**string name;**

**char mod;**

**char grade;**

**public:**

**void addinput()**

**{**

**ofstream file("studentrecord.txt",ios::app);**

**if (file.is\_open())**

**{**

**cout<<"Enter your full name "<<endl;**

**getline(cin,name);**

**file<<"Name : "<<name<<endl;**

**cout<<"Enter your roll number "<<endl;**

**cin>>roll;**

**file<<"Roll no : "<<roll<<endl;**

**cin.ignore();**

**cout<<"Enter marks of Subject 1 out of 100 "<<endl;**

**cin>>mark1;**

**if (mark1>=0 && mark1<=100)**

**{**

**file<<"Mark1 : "<<mark1<<endl;**

**}**

**else**

**{**

**file << "Mark1 : INVALID MARK" << endl;**

**cout<<"INVALID MARK";**

**}**

**cout<<"Enter marks of Subject 2 out of 100 "<<endl;**

**cin>>mark2;**

**if (mark2>=0 && mark2<=100)**

**{**

**file<<"Mark 2 : "<<mark2<<endl;**

**}**

**else**

**{**

**file << "Mark2 : INVALID MARK" << endl;**

**cout<<"INVALID MARK";**

**}**

**cout<<"Enter marks of Subject 3 out of 100"<<endl;**

**cin>>mark3;**

**if (mark3>=0 && mark3<=100)**

**{**

**file<<"Mark 3 : "<<mark3<<endl;**

**}**

**else**

**{**

**file << "Mark3 : INVALID MARK" << endl;**

**cout<<"INVALID MARKS";**

**}**

**}**

**else**

**{**

**cout<<"File not open";**

**}**

**}**

**void modify()**

**{**

**cout <<"DO YOU WISH TO MODIFY RECORDS (y/n)"<<endl;**

**cin>>mod;**

**cin.ignore();**

**if (mod=='Y' || mod=='y')**

**{**

**string searchh;**

**string line;**

**cout<<"enter the name of the student you want to check: ";**

**getline(cin,searchh);**

**ifstream infile("studentrecord.txt");**

**ofstream temp("temp.txt");**

**if (!infile.is\_open()) {**

**cout << "File cannot be opened!" << endl;**

**return;**

**}**

**bool found = false;**

**while (getline(infile, line))**

**{**

**if (line.find("Name : " + searchh) != string::npos)**

**{**

**found=true;**

**temp << line << endl;**

**getline(infile, line); temp << line << endl;**

**cout << "Enter new marks for " << searchh << ":\n";**

**cout << "Subject 1: ";**

**cin >> mark1;**

**if (mark1 >= 0 && mark1 <= 100)**

**temp << "Mark1 : " << mark1 << endl;**

**else**

**temp << "Mark1 : INVALID MARK" << endl;**

**cout << "Subject 2: ";**

**cin >> mark2;**

**if (mark2 >= 0 && mark2 <= 100)**

**temp << "Mark2 : " << mark2 << endl;**

**else**

**temp << "Mark2 : INVALID MARK" << endl;**

**cout << "Subject 3: ";**

**cin >> mark3;**

**if (mark3>= 0 && mark3 <= 100)**

**temp << "Mark3 : " << mark3 << endl;**

**else**

**temp << "Mark3 : INVALID MARK" << endl;**

**temp << "Mark1 : " << mark1 << endl;**

**temp << "Mark2 : " << mark2 << endl;**

**temp << "Mark3 : " << mark3 << endl;**

**getline(infile, line);**

**getline(infile, line);**

**getline(infile, line);**

**}**

**else**

**{**

**temp<<line<<endl;**

**}**

**}**

**if (!found) {**

**cout << "Student not found!" << endl;**

**}**

**infile.close();**

**temp.close();**

**remove("studentrecord.txt");**

**rename("temp.txt", "studentrecord.txt");**

**}**

**else {**

**cout<<"END"<<endl;**

**}**

**}**

**void display()**

**{**

**ifstream openfile("studentrecord.txt");**

**string lines;**

**cout << "records : "<<endl;**

**while (getline(openfile, lines)) {**

**cout << lines << endl;**

**}**

**openfile.close();**

**}**

**};**

**int main(){**

**student s;**

**s.addinput();**

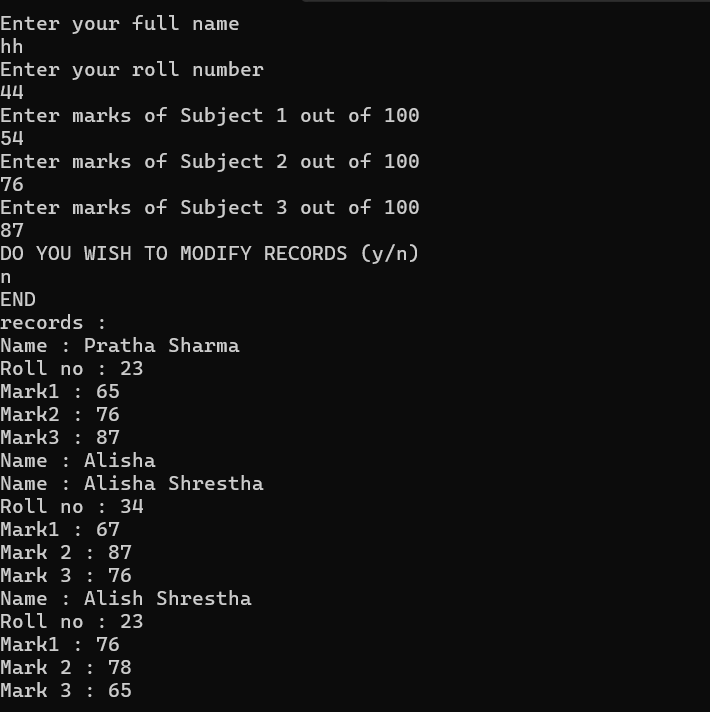
**s.modify();**

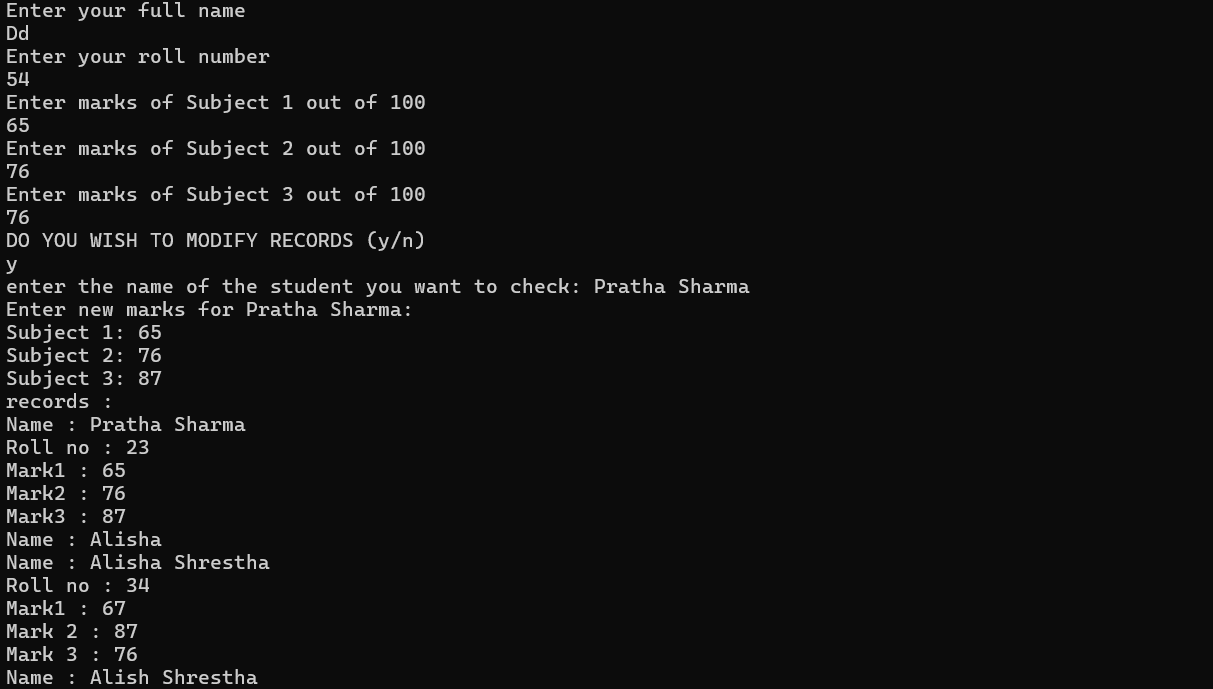
**s.display();**

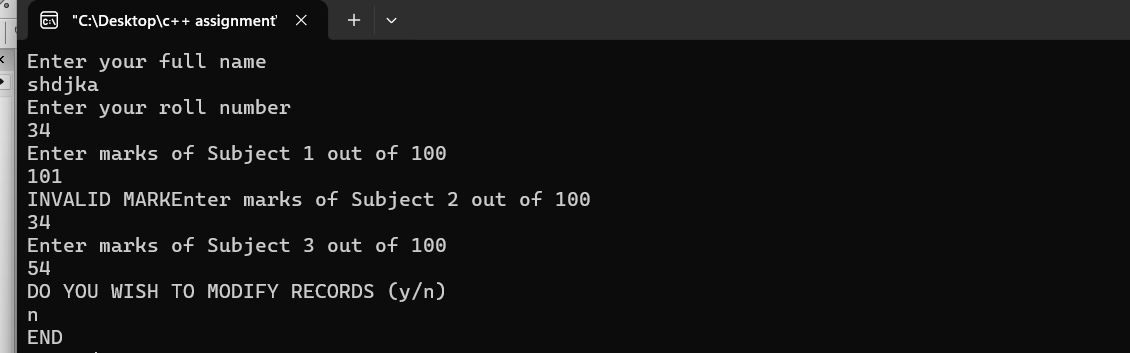
**return 0;**

**}**

**OUTPUT:**

****

****

****