**Lab File**

**Object Oriented Programming Using C++**

**(ES 203)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



Submitted to: Submitted by:

Dr Vandana Bhatia Shaina Mehta

Ast. Professor A2305219268

CSE Department, ASET B.tech. C.S.E.

3CSE-4Y

Amity School Of Engineering and technology

Amity University Uttar Pradesh

Noida -201301

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exp No | Assignment  Category | Code | Name of Experiment | Date of Allotment | Date of Evaluation | Max  Marks | Marks  Obtained | Faculty  Sign |
| 1 | Mandatory  Experiment |  | Write a program to check number palindrome. | 16-07-2020 | 30-10-2020 |  |  |  |
| 2 |  | Write a program to show the effect of call by value and call by reference in functions. | 23-07-2020 | 30-10-2020 |  |  |  |
| 3 |  | Write a program to perform following operations on matrix using functions and switch  case:  (a) Addition (b)subtraction (c) multiplication (d) transpose | 30-07-2020 | 30-10-2020 |  |  |  |
| 4 |  | Define a class Shape whose attributes are radius, length and width calculate the perimeter  of the rectangle and circle. | 6-08-2020 | 30-10-2020 |  |  |  |
| 5 |  | Define the methods of the class inside or outside the class declaration. Use the concept of  scope resolution operator. | 13-08-2020 | 30-10-2020 |  |  |  |
| 6 |  | Write a program to develop the salary slip and display result by using constructors. | 20-08-2020 | 30-10-2020 |  |  |  |
| 7 |  | Write a program to find the biggest of three numbers using friend function | 3-09-2020 | 30-10-2020 |  |  |  |
| 8 |  | Write a C++ program to implement static data members and static member functions. | 10-09-2020 | 30-10-2020 |  |  |  |
| 9 |  | Write a C++ program to call base class constructors in the following forms of inheritance. a) Single Inheritance b) Multiple Inheritance c) Multi level Inheritance d) Hierarchical  Inheritance. | 17-09-2020 | 30-10-2020 |  |  |  |
| 10 |  | Write a program to find the area of circle, rectangle and triangle by Function overloading  concept. | 24-09-2020 | 30-10-2020 |  |  |  |
| 11 |  | Write a C++ program that overloads the unary ++ operator to increment each element of  the given one -dimensional array by ‘1’. | 1-10-2020 | 30-10-2020 |  |  |  |
| 12 |  | Write a program to calculate the total mark of a student using the concept of virtual base  class. | 8-10-2020 | 30-10-2020 |  |  |  |
| 13 |  | Write a C++ program that to perform various operations on strings handling using string  class. | 15-10-2020 | 30-10-2020 |  |  |  |
| 14 |  | Write a C++ program to display the contents of a text file. | 22-10-2020 | 30-10-2020 |  |  |  |
| 15 |  | Write a program to implement stack functions using templates. | 29-10-2020 | 30-10-2020 |  |  |  |
|  | Viva | Viva |  |  |  |  |  |  |

**Q1) Write a program to check number palindrome.**

**Ans)**

**Code:**

//Addition.cpp

#include <iostream>

using namespace std;

int main()

{

int n;

int a,r,d=0;

cout<<"enter a number";

cin>>n;

a=n;

while(a>0)

{

r=a%10;

d=d\*10+r;

a=a/10;

}

if(n==d)

{

cout<<"It is a palindrome";

}

else

{

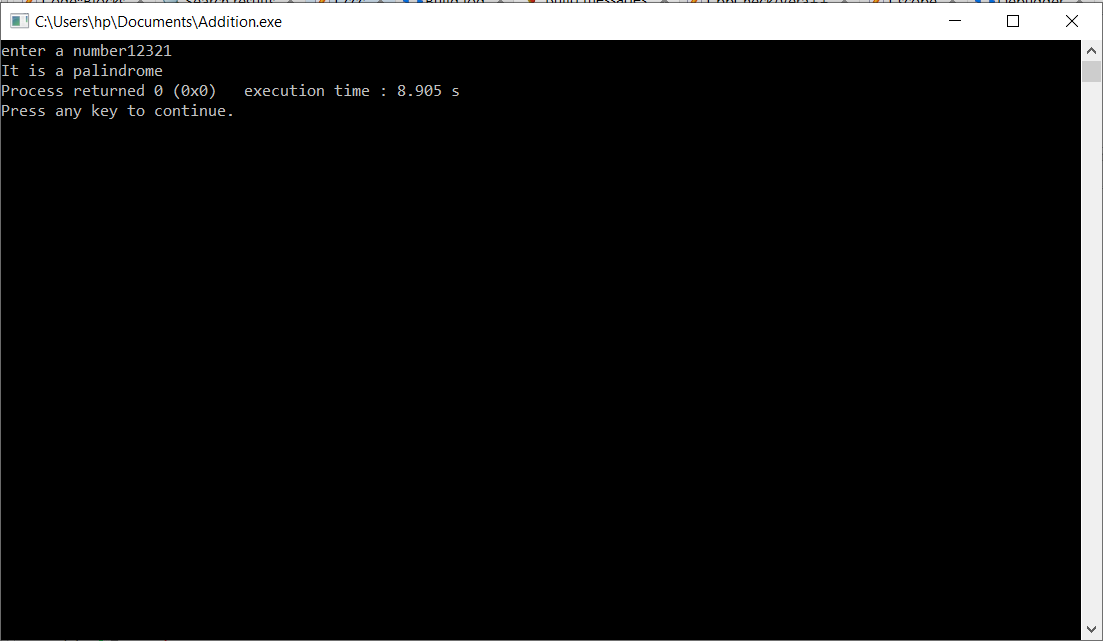
cout<<"It is not a palindrome";

}

return 0;

}

**Output:**



**Q2) Write a program to show the effect of call by value and call by reference in functions.**

**Ans)**

**Code:**

//CBVCBR.cpp

#include <iostream>

using namespace std;

int add(int a,int b)//call By Value

{

return a+b;

}

int add(int \*a,int \*b)//Call By Reference

{

return \*a+\*b;

}

int main()

{

int a,b,c;

cout<<"enter the first number";

cin>>a;

cout<<"enter the second number";

cin>>b;

c=add(a,b);//Call By Value

cout<<a<<"+"<<b<<"="<<c<<endl;

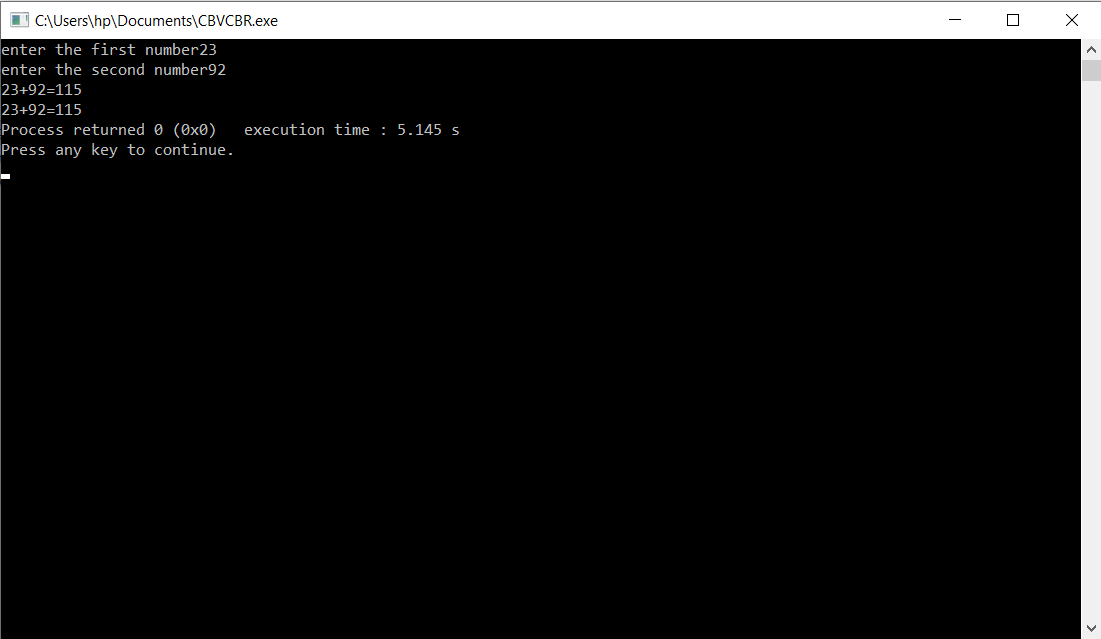
c=add(&a,&b);//Call By Reference

cout<<a<<"+"<<b<<"="<<c;

return 0;

}

**Output:**



**Q3) Write a program to perform following operations on matrix using functions and switch case:**

**(a) Addition (b)subtraction (c) multiplication (d) transpose.**

**Ans)**

**Code:**

//Matrix Calculator.cpp

#include<iostream>

#include<process.h>

using namespace std;

void addmat(int a[][100],int r1,int c1,int b[][100],int c[][100])

{

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

\*(\*(c+i)+j)=\*(\*(b+i)+j)+\*(\*(a+i)+j);

}

}

}

void submat(int a[][100],int r1,int c1,int b[][100],int c[][100])

{

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

\*(\*(c+i)+j)=\*(\*(b+i)+j)-\*(\*(a+i)+j);

}

}

}

void multimat(int a[][100],int r1,int b[][100],int c2,int c[][100],int p)

{

for(int i=0;i<r1;i++)

{

for(int j=0;j<c2;j++)

{

int sum=0;

for(int k=0;k<p;k++)

{

int d=a[i][k]\*b[k][j];

sum=sum+d;

}

c[i][j]=sum;

}

}

}

void transpose(int a[][100],int r,int c,int b[][100])

{

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

{

for(int j=0;j<c;j++)

{

b[i][j]=a[j][i];

}

}

}

int main()

{

int a[100][100],b[100][100],result[100][100];

int r1,r2,c1,c2;

int choice;

while(1)

{

cout<<"Matrix Calculation"<<endl;

cout<<"1.Addition"<<endl;

cout<<"2.Subtraction"<<endl;

cout<<"3.Multiplication"<<endl;

cout<<"4.Transpose"<<endl;

cout<<"5.Exit"<<endl;

cout<<"Enter your choice: ";

cin>>choice;

switch(choice)

{

case 1: cout<<"Enter the number of rows of matrix 1: ";

cin>>r1;

cout<<"Enter the number of columns of matrix 1: ";

cin>>c1;

cout<<"Enter the number of rows of matrix 2: ";

cin>>r2;

cout<<"Enter the number of columns of matrix 2: ";

cin>>c2;

if(r1!=r2 || c1!=c2)

{

cout<<"The matrix addition is not possible."<<endl;

continue;

}

cout<<"Enter the elements of matrix 1: ";

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cin>>a[i][j];

}

}

cout<<"Enter the elements of matrix 2: ";

for(int i=0;i<r2;i++)

{

for(int j=0;j<c2;j++)

{

cin>>b[i][j];

}

}

cout<<"The matrix sum is: "<<endl;

addmat(a,r1,c1,b,result);

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cout<<result[i][j]<<" ";

}

cout<<endl;

}

break;

case 2: cout<<"Enter the number of rows of matrix 1: ";

cin>>r1;

cout<<"Enter the number of columns of matrix 1: ";

cin>>c1;

cout<<"Enter the number of rows of matrix 2: ";

cin>>r2;

cout<<"Enter the number of columns of matrix 2: ";

cin>>c2;

if(r1!=r2 || c1!=c2)

{

cout<<"The matrix subtraction is not possible."<<endl;

continue;

}

cout<<"Enter the elements of matrix 1: ";

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cin>>a[i][j];

}

}

cout<<"Enter the elements of matrix 2: ";

for(int i=0;i<r2;i++)

{

for(int j=0;j<c2;j++)

{

cin>>b[i][j];

}

}

cout<<"The matrix difference is: "<<endl;

submat(a,r1,c1,b,result);

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cout<<result[i][j]<<" ";

}

cout<<endl;

}

break;

case 3: cout<<"Enter the number of rows of matrix 1: ";

cin>>r1;

cout<<"Enter the number of columns of matrix 1: ";

cin>>c1;

cout<<"Enter the number of rows of matrix 2: ";

cin>>r2;

cout<<"Enter the number of columns of matrix 2: ";

cin>>c2;

if(c1!=r2)

{

cout<<"The matrix multiplication is not possible."<<endl;

continue;

}

cout<<"Enter the elements of matrix 1: ";

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cin>>a[i][j];

}

}

multimat(a,r1,b,c2,result,c1);

cout<<"Enter the elements of matrix 2: ";

for(int i=0;i<r2;i++)

{

for(int j=0;j<c2;j++)

{

cin>>b[i][j];

}

}

cout<<"The matrix multiplication is: "<<endl;

multimat(a,r1,b,c2,result,c1);

for(int i=0;i<r1;i++)

{

for(int j=0;j<c2;j++)

{

cout<<result[i][j]<<" ";

}

cout<<endl;

}

break;

case 4: cout<<"Enter the number of rows of matrix : ";

cin>>r1;

cout<<"Enter the number of columns of matrix : ";

cin>>c1;

cout<<"Enter the elements of matrix : ";

if(r1!=c1)

{

cout<<"The Transpose of a matrix is not possible"<<endl;

continue;

}

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cin>>a[i][j];

}

}

cout<<"The transpose is: "<<endl;

transpose(a,r1,c1,b);

for(int i=0;i<r1;i++)

{

for(int j=0;j<c1;j++)

{

cout<<b[i][j]<<" ";

}

cout<<endl;

}

break;

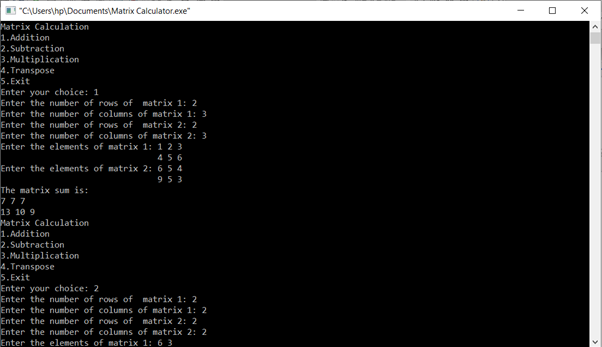
case 5: exit(0);

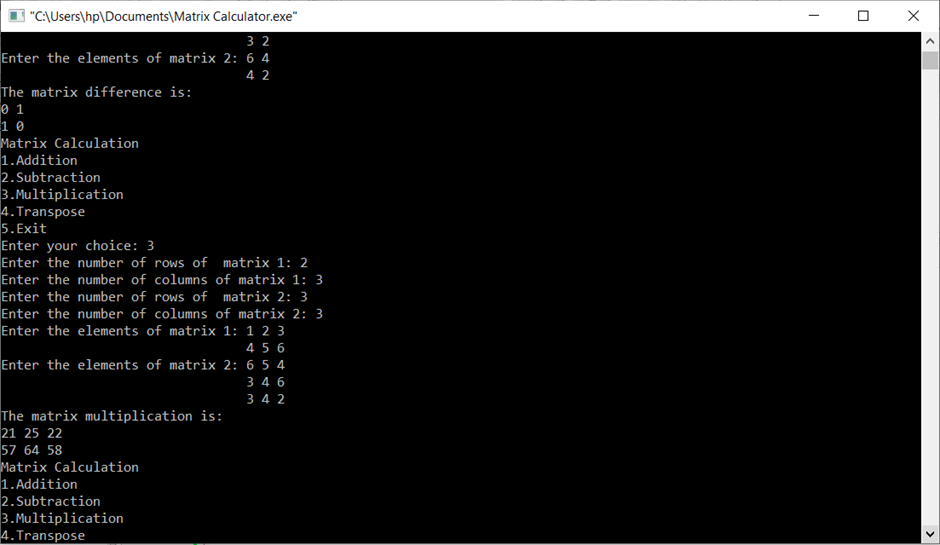
}

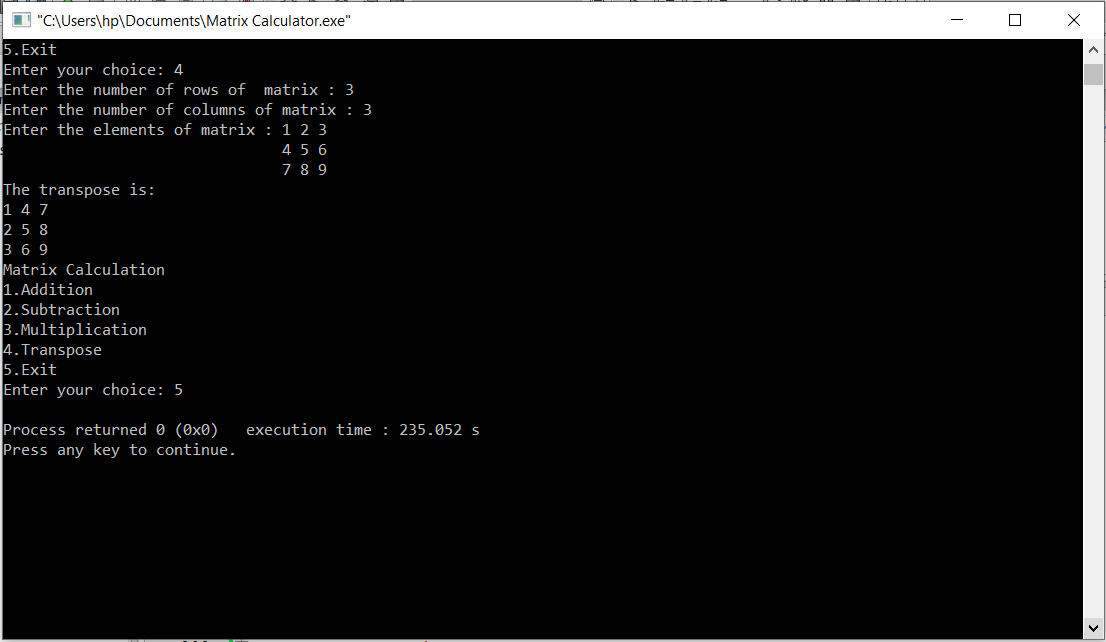
}

}

**Output:**







**Q4) Define a class Shape whose attributes are radius, length and width calculate the perimeter of the rectangle and circle.**

**Ans)**

**Code:**

//shape.cpp

#define pi 3.14

template <typename T>

class shape

{

T radius;

T length;

T breadth;

public:

shape(T radius,T length,T breadth)

{

this->radius=radius;

this->length=length;

this->breadth=breadth;

}

T cperimeter()

{

return 2\*pi\*radius;

}

T rperimeter()

{

return 2\*(length+breadth);

}

~shape()

{}

};

//shapeuse.cpp

#include <iostream>

using namespace std;

#include "shape.cpp"

int main()

{

float r,l,b;

cout<<"enter the radius of the circle";

cin>>r;

cout<<"enter the length of the rectangle";

cin>>l;

cout<<"enter the breadth of the rectangle";

cin>>b;

shape <float> s(r,l,b);

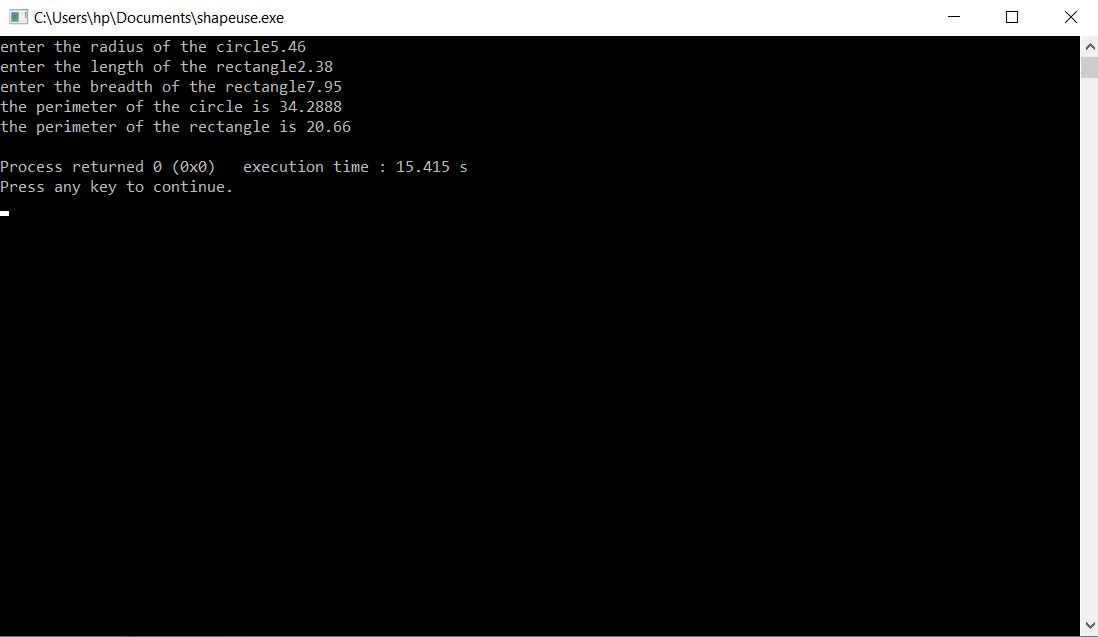
cout<<"the perimeter of the circle is"<<" "<<s.cperimeter()<<endl;

cout<<"the perimeter of the rectangle is"<<" "<<s.rperimeter()<<endl;

return 0;

}

**Output:**



**Q5) Define the methods of the class inside or outside the class declaration. Use the concept of scope resolution operator.**

**Ans)**

**Code:**

//Scope Resolution Operator.cpp

#include<iostream>

using namespace std;

class Time

{

int hours, minutes, seconds;

public:

Time(): hours(00), minutes(00), seconds(00)//Constructor overloading

{}//constructor1

Time(int h,int m,int s): hours(h), minutes(m), seconds(s)//Constructor overloading

{}//constructor2

void addtime(Time,Time);

void showtime()

{

cout<<"The time is"<<hours<<":"<<minutes<<":"<<seconds<<endl;

}

};

void Time::addtime(Time t1, Time t2)//Passing objects of the class Time as the function arguments

{

seconds=t1.seconds+t2.seconds;

while(seconds>=60)

{

seconds-=60;

minutes++;

}

minutes+=t1.minutes+t2.minutes;

while(minutes>=60)

{

minutes-=60;

hours++;

}

hours+=t1.hours+t2.hours;

}

int main()

{

Time t1(45,23,89);//calling constructor 2

Time t2(15,114,43);//calling constructor 2

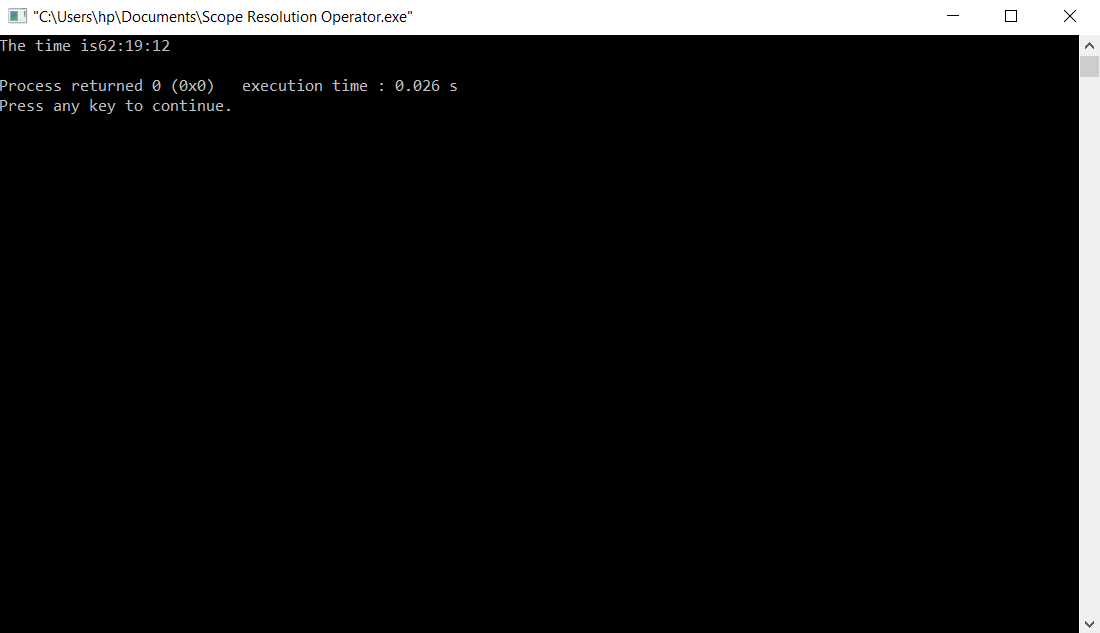
Time t3;//calling constructor 1

t3.addtime(t1,t2);

t3.showtime();

}

**Output:**



**Q6) Write a program to develop the salary slip and display result by using constructors.**

**Ans)**

**Code:**

//Salary Scale.cpp

#include<iostream>

#include <string>

using namespace std;

class employee

{

string name;

string ss;

float basic;

string dept;

string wf;

float salary;

public:

int id;

employee(): name("Unknown Employee"), ss("Low"), basic(12000.00), dept("Nil"), wf("All Days"), salary(0.00), id(0)

{}

void inputdetails()

{

cout<<"Enter the name: "<<endl;

cin>>name;

cout<<"Enter the salaryscale: "<<endl;

cin>>ss;

cout<<"Enter the department: "<<endl;

cin>>dept;

}

string weekoff()

{

if(id%2==0)

{

wf="Saturday";

}

else

{

wf="Sunday";

}

return wf;

}

float computesalary()

{

float da,hra;

if(ss=="High")

{

da=basic\*0.25;

hra=basic\*1.20;

salary=da+hra+basic;

}

else if(ss=="Medium")

{

da=basic\*0.15;

hra=basic\*1;

salary=da+hra+basic;

}

else

{

salary=basic;

}

return salary;

}

void displaydetails()

{

cout<<"Employee ID: "<<id<<endl;

cout<<"Name: "<<name<<endl;

cout<<"Department: "<<dept<<endl;

cout<<"Salaryscale: "<<ss<<endl;

cout<<"Weekoff: "<<weekoff()<<endl;

cout<<"Salary: "<<computesalary()<<endl;

}

};

int main()

{

employee e[100];

int entries;

cout<<"Enter the no of entries: "<<endl;

cin>>entries;

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

{

cout<<"Enter the data:"<<endl;

e[i].id=i+1;

e[i].inputdetails();

}

cout<<"The employee details are:"<<endl;

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

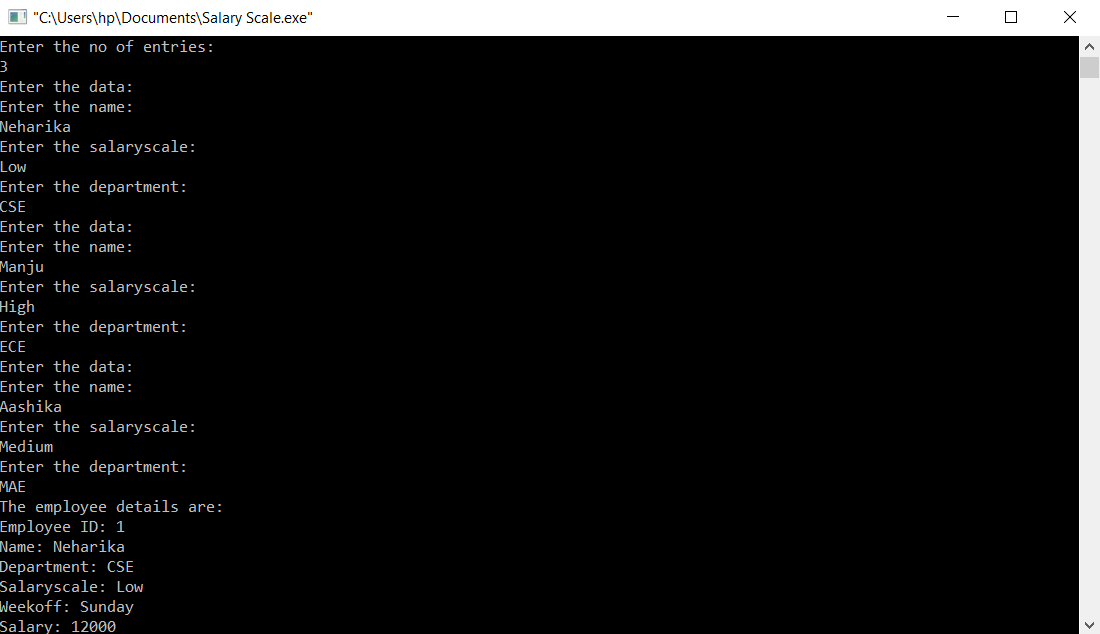
{

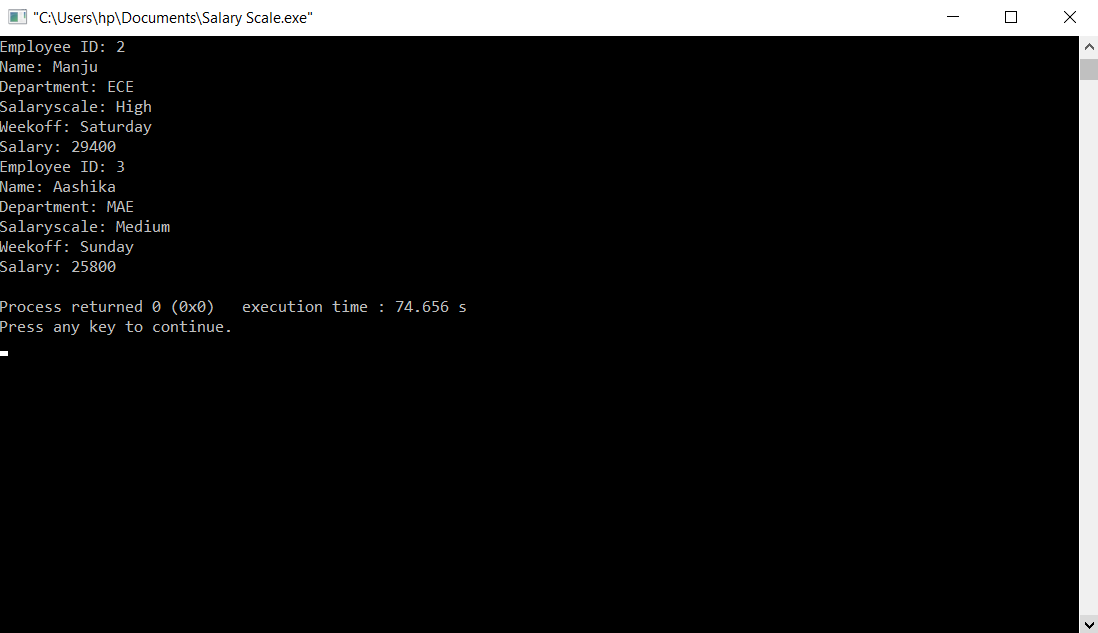
e[i].displaydetails();

}

}

**Output:**





**Q7) Write a program to find the biggest of three numbers using friend function**

**Ans)**

**Code:**

//Greatest Friend.cpp

#include <iostream>

using namespace std;

class highest

{

int A,B,C;

public:

highest(int a, int b,int c): A(a), B(b), C(c)

{}

friend int greatest (highest);

};

int greatest(highest h)

{

return ((h.A>=h.B && h.A>=h.C)? h.A: ((h.B>=h.A && h.B>=h.C)? h.B: h.C));

}

int main()

{

int a,b,c;

cout<<"Enter the first number: ";

cin>>a;

cout<<"Enter the second number: ";

cin>>b;

cout<<"Enter the third number: ";

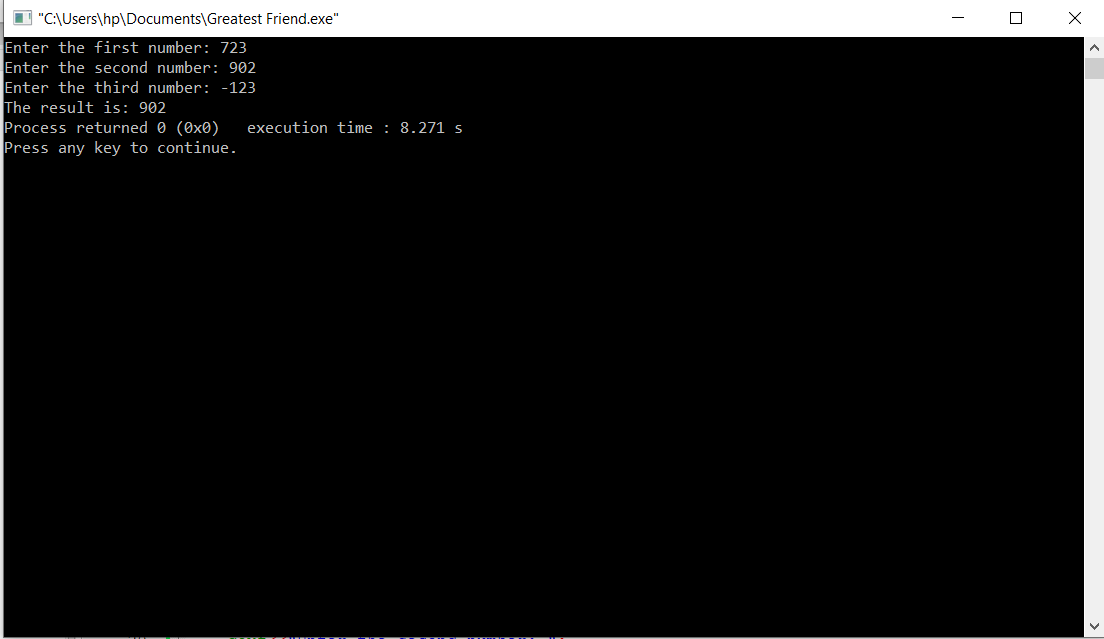
cin>>c;

highest h(a,b,c);

cout<<"The result is: "<<greatest(h);

return 0;

**Output:**



**Q8) Write a C++ program to implement static data members and static member functions.**

**Ans)**

**Code:**

//SDMSMF.cpp

#include<iostream>

#include<string>

using namespace std;

class Student

{

string Name;

string RollNo;

int Mathematics,Physics,Chemistry,Result;

public:

//Static Data Member

static int total;

Student()

{

Name="NoName";

RollNo="S00";

Mathematics=0;

Physics=0;

Chemistry=0;

Result=0;

}

void Input()

{

cout<<"Enter the Name: ";

cin>>Name;

cout<<"Enter the Roll No: ";

cin>>RollNo;

cout<<"Enter the marks of Mathematics: ";

cin>>Mathematics;

cout<<"Enter the marks of Physics: ";

cin>>Physics;

cout<<"Enter the marks of Chemistry: ";

cin>>Chemistry;

}

//Static Member Function

static void ShowTotal()

{

cout<<"The total no of students are: "<<total<<endl;

}

void Display()

{

Result=Mathematics+Physics+Chemistry;

cout<<"Name: "<<Name<<endl;

cout<<"Roll No: "<<RollNo<<endl;

cout<<"Mathematics: "<<Mathematics<<endl;

cout<<"Physics: "<<Physics<<endl;

cout<<"Chemistry: "<<Chemistry<<endl;

cout<<"Total Marks: "<<Result<<endl;

}

~Student()

{}

};

//Initialization of Static Function

int Student::total=0;

int main()

{

Student s[100];

int n;

cout<<"Enter the no of student data that you want to enter: ";

cin>>n;

cout<<"Enter the student data:"<<endl;

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

{

s[i].Input();

//Declaration of Static Data Member in the main() Function

Student::total++;

}

cout<<"The student record is:"<<endl;

//Calling of Static Function

Student::ShowTotal();

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

{

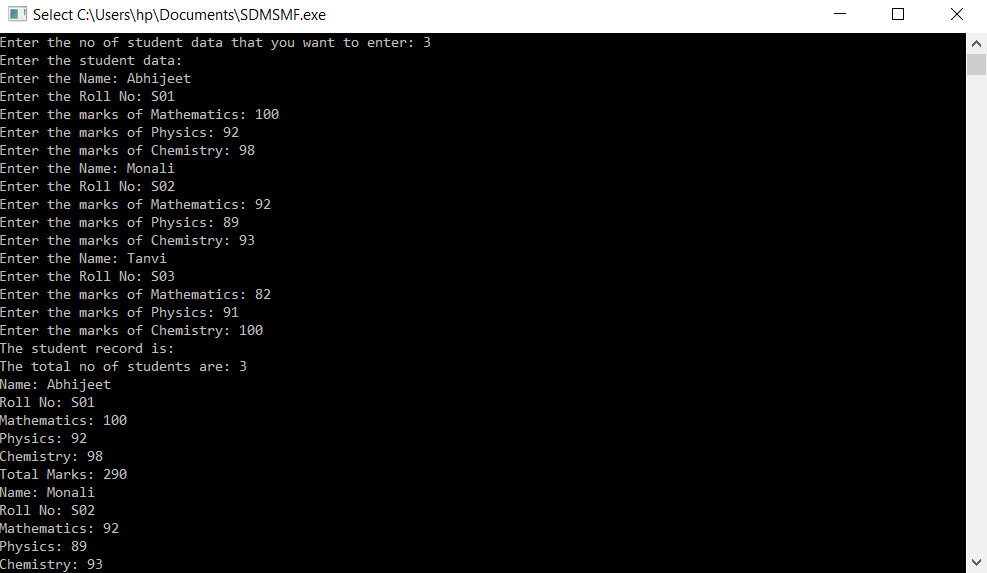
s[i].Display();

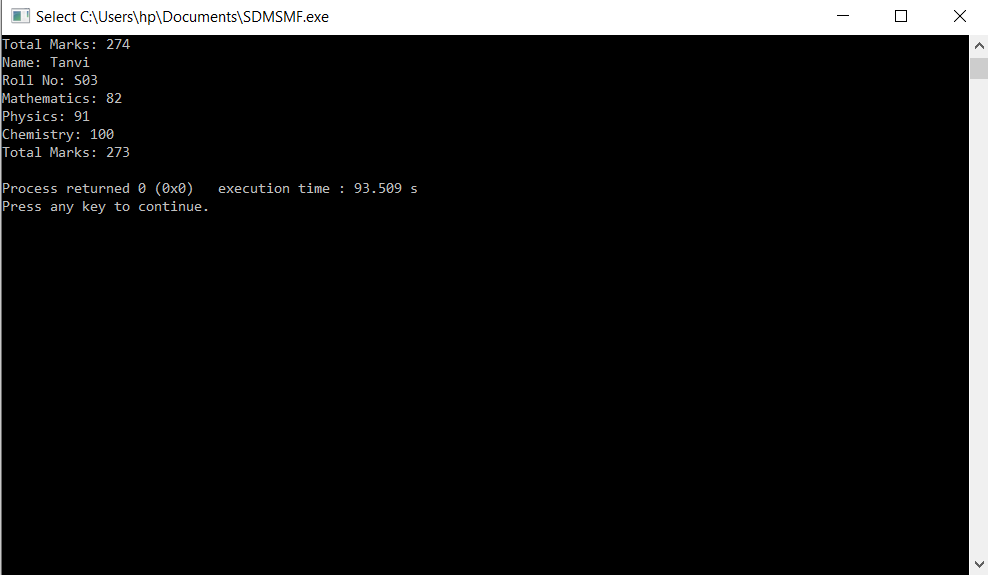
}

return 0;

}

**Output:**





**Q9) Write a C++ program to call base class constructors in the following forms of inheritance.**

**a) Single Inheritance b) Multiple Inheritance c) Multi level Inheritance d) Hierarchical**

**Inheritance**

**Ans)**

**(a)**

**Code:**

//Single Inheritance.cpp

#include<iostream>

using namespace std;

class Base

{

public:

Base()

{

cout<<"Base Class Constructor is called."<<endl;

}

};

class Derived: public Base

{

public:

Derived()

{

cout<<"Derived Class Constructor is called."<<endl;

}

};

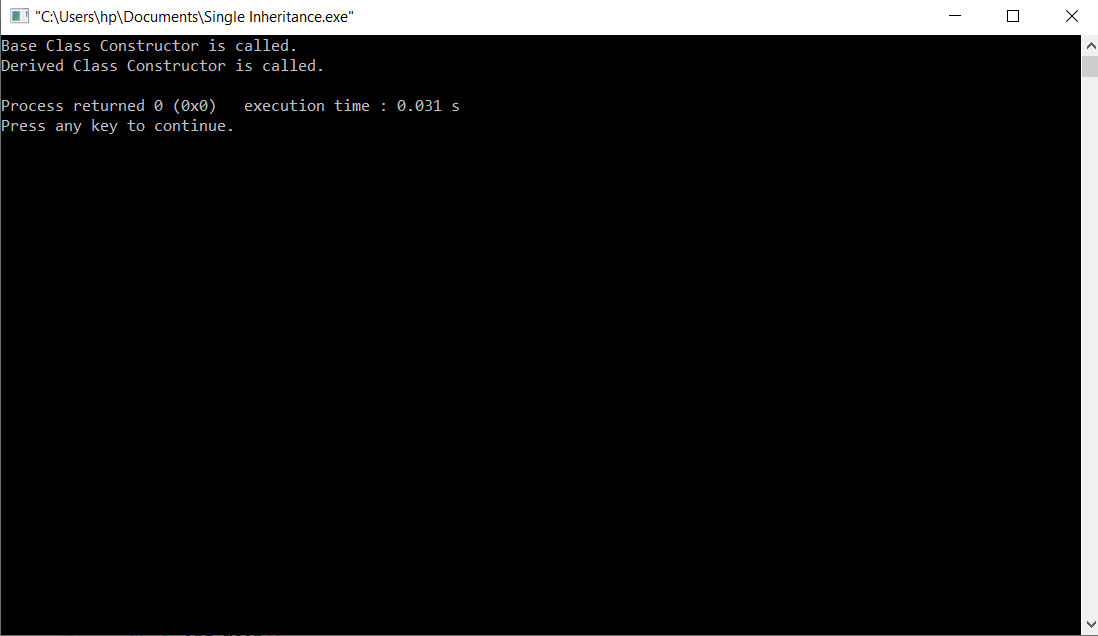
int main()

{

Derived d;

}

**Output:**



**(b)**

**Code:**

//Multiple Inheritance.cpp

#include<iostream>

using namespace std;

class Base1

{

public:

Base1()

{

cout<<"Base1 Class Constructor is called."<<endl;

}

};

class Base2

{

public:

Base2()

{

cout<<"Base2 Class Constructor is called."<<endl;

}

};

class Derived: public Base1, public Base2

{

public:

Derived()

{

cout<<"Derived Class Constructor is called."<<endl;

}

};

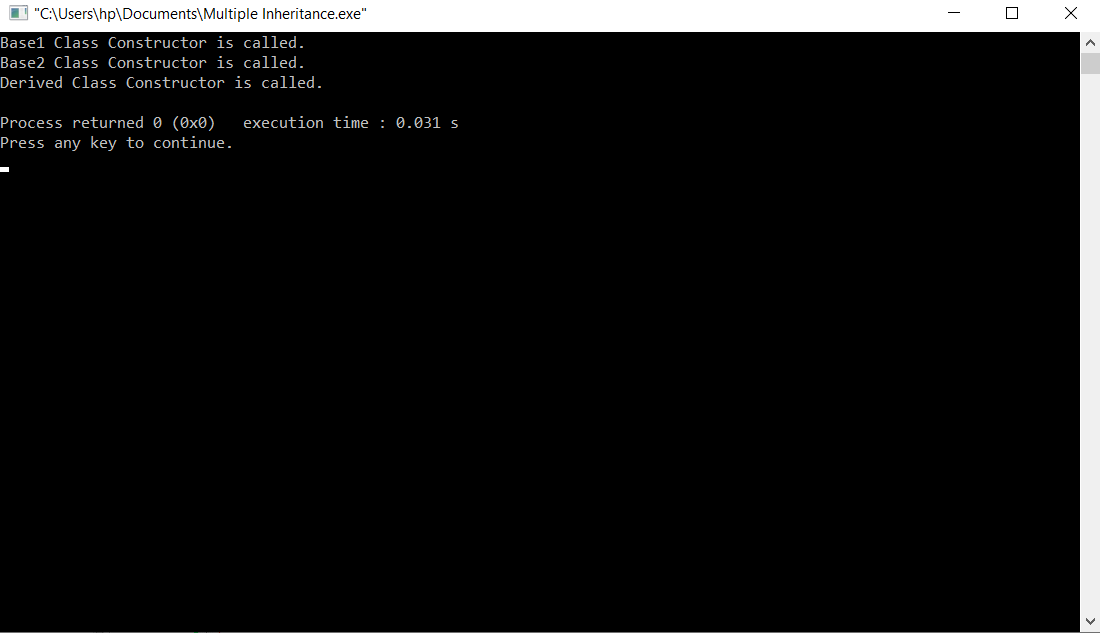
int main()

{

Derived d;

}

**Output:**



**(c)**

**Code:**

//Multilevel Inheritance.cpp

#include<iostream>

using namespace std;

class Base

{

public:

Base()

{

cout<<"Base Class Constructor is called."<<endl;

}

};

class Derived\_Base: public Base

{

public:

Derived\_Base()

{

cout<<"Derived\_Base Class Constructor is called."<<endl;

}

};

class Derived: public Derived\_Base

{

public:

Derived()

{

cout<<"Derived Class Constructor is called."<<endl;

}

};

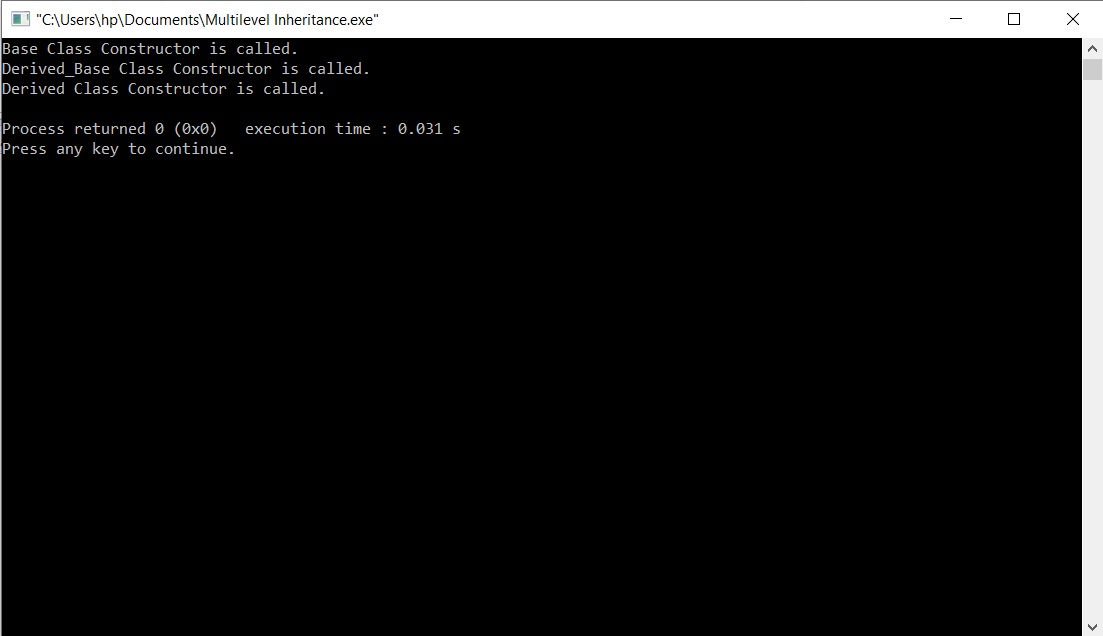
int main()

{

Derived d;

}

**Output:**



**(d)**

**Code:**

//Hierarchical Inheritance.cpp

#include<iostream>

using namespace std;

class Base

{

public:

Base()

{

cout<<"Base Class Constructor is called."<<endl;

}

};

class Derived1: public Base

{

public:

Derived1()

{

cout<<"Derived1 Class Constructor is called."<<endl;

}

};

class Derived2: public Base

{

public:

Derived2()

{

cout<<"Derived2 Class Constructor is called."<<endl;

}

};

int main()

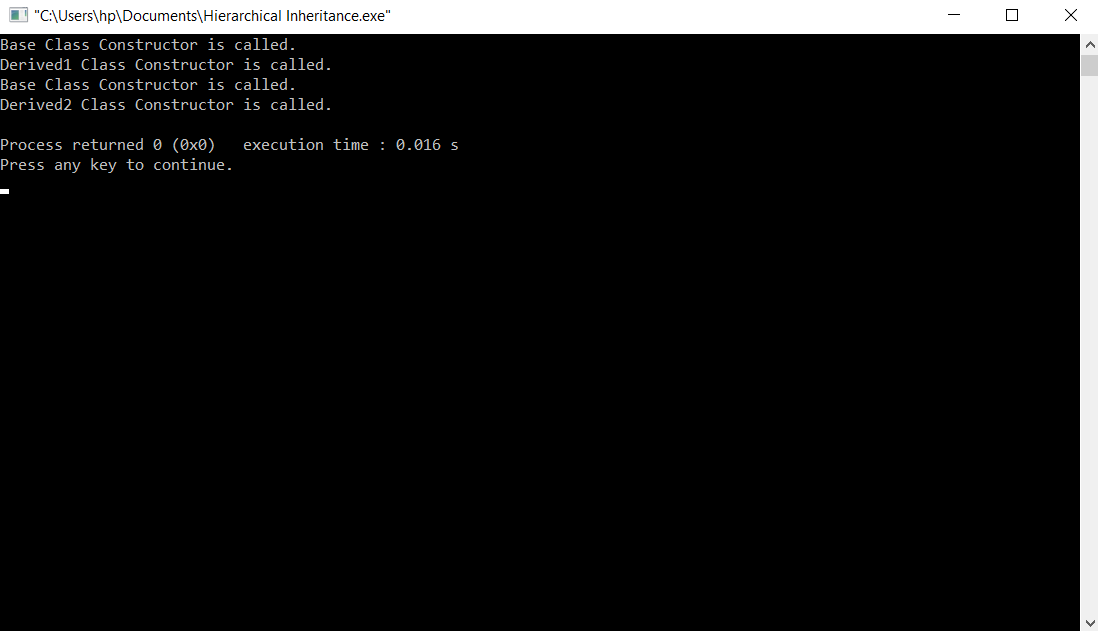
{

Derived1 d1;

Derived2 d2;

}

**Output:**



**Q10) Write a program to find the area of circle, rectangle and triangle by Function overloading concept.**

**Ans)**

**Code:**

//Area.cpp

#include<iostream>

#include<cmath>

#define pi 3.14

float area(float l,float b)

{

return l\*b;

}

float area(float h,float b,float x)

{

return h\*b\*x;

}

float area(float r)

{

return pi\*r\*r;

}

using namespace std;

int main()

{

float x=0.5;

float r,l,b,h,a;

int ch;

while(1)

{

cout<<"Main Menu"<<endl;

cout<<"1.Area of the rectangle."<<endl;

cout<<"2.Area of the triangle."<<endl;

cout<<"3.Area of the circle."<<endl;

cout<<"4.Exit."<<endl;

cout<<"Enter your choice: ";

cin>>ch;

switch(ch)

{

case 1:cout<<"Enter the length of the rectangle: ";

cin>>l;

cout<<"Enter the breadth of the rectangle: ";

cin>>b;

a=area(l,b);

cout<<"The area of the rectangle is: "<<a<<endl;

break;

case 2:cout<<"Enter the height of the triangle: ";

cin>>h;

cout<<"Enter the breadth of the triangle: ";

cin>>b;

a=area(h,b,x);

cout<<"The area of the triangle is: "<<a<<endl;

break;

case 3:cout<<"Enter the radius of the circle: ";

cin>>r;

a=area(r);

cout<<"The area of the circle is: "<<a<<endl;

break;

case 4:exit(0);

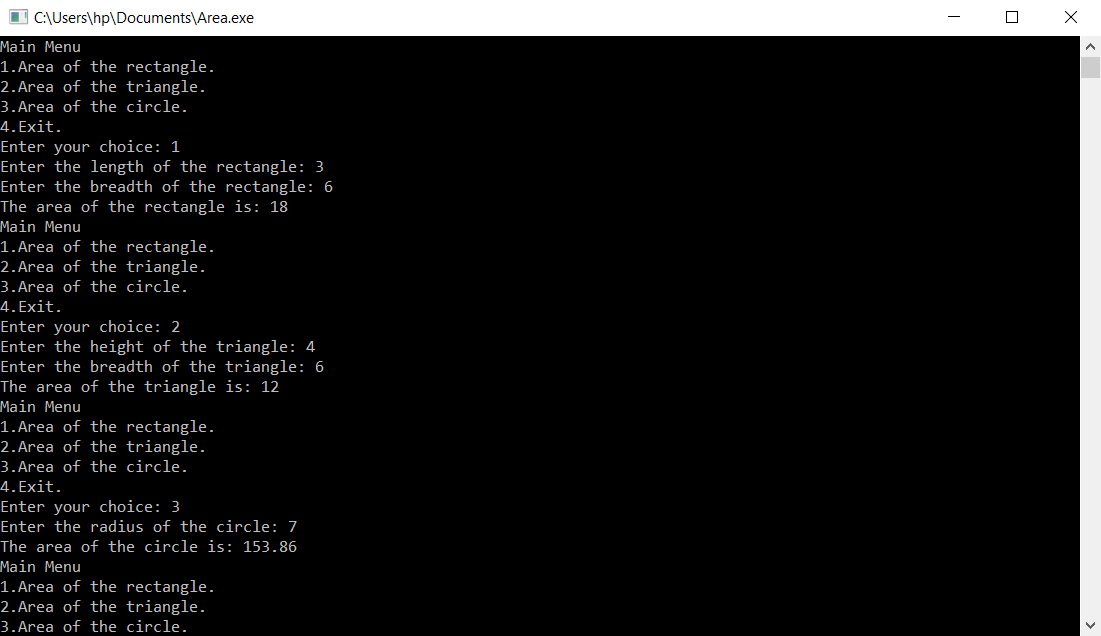
break;

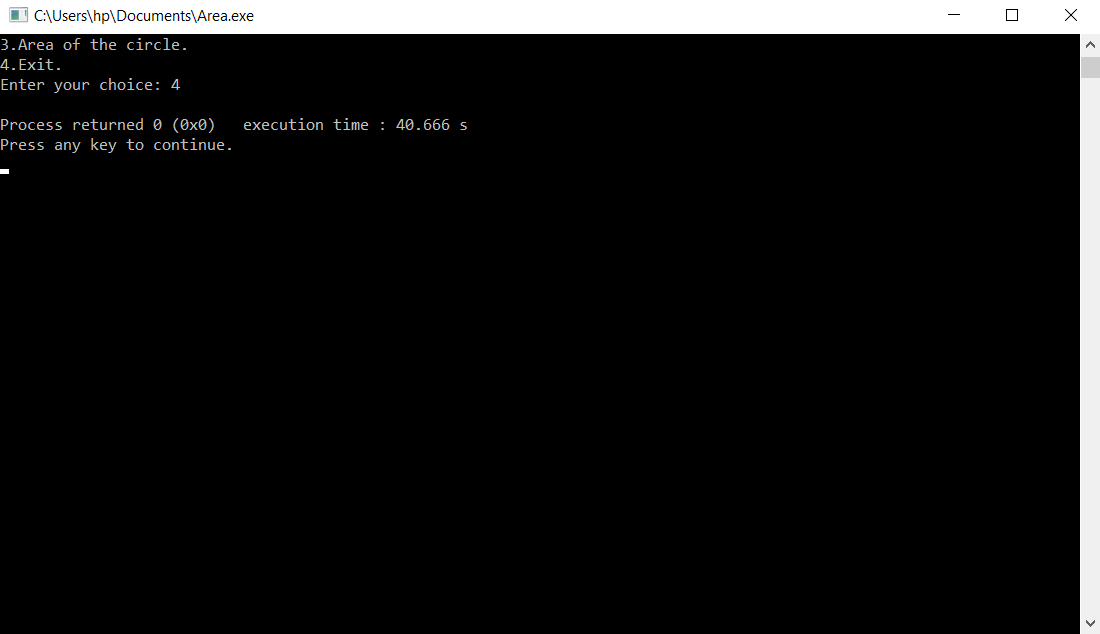
}

}

}

**Output:**





**Q11) Write a C++ program that overloads the unary ++ operator to increment each element of the given one - dimensional array by ‘1’.**

**Ans)**

**Code:**

//OOArray.cpp

#include<iostream>

using namespace std;

class uniary\_operator

{

int a[10];

public:

uniary\_operator()

{

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

{

this->a[i]=0;

}

}

uniary\_operator(int a[])

{

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

{

this->a[i]=a[i];

}

}

uniary\_operator operator++()

{

uniary\_operator ui(a);

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

{

ui.a[i]=++a[i];

}

return ui;

}

uniary\_operator operator--()

{

uniary\_operator ui(a);

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

{

ui.a[i]=--a[i];

}

return ui;

}

uniary\_operator operator++(int)

{

uniary\_operator ui(a);

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

{

ui.a[i]=a[i]++;

}

return ui;

}

uniary\_operator operator--(int)

{

uniary\_operator ui(a);

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

{

ui.a[i]=a[i]--;

}

return ui;

}

void print()

{

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

{

cout<<a[i]<<" ";

}

cout<<endl;

}

};

int main()

{

int a[10]={84,23,13,23,76,40,93,21,36,84};

uniary\_operator u1(a);

cout<<"The original array is: ";

u1.print();

cout<<"The pre increment of array is: "<<endl;

uniary\_operator u2;

u2=++u1;

cout<<"In the result: ";

u2.print();

cout<<"In the memory: ";

u1.print();

cout<<"The post increment of array is: "<<endl;

uniary\_operator u3;

u3=u1++;

cout<<"In the result: ";

u3.print();

cout<<"In the memory: ";

u1.print();

cout<<"The pre decrement of array is: "<<endl;

uniary\_operator u4;

u4=--u1;

cout<<"In the result: ";

u4.print();

cout<<"In the memory: ";

u1.print();

cout<<"The post decrement of array is: "<<endl;

uniary\_operator u5;

u5=u1--;

cout<<"In the result: ";

u5.print();

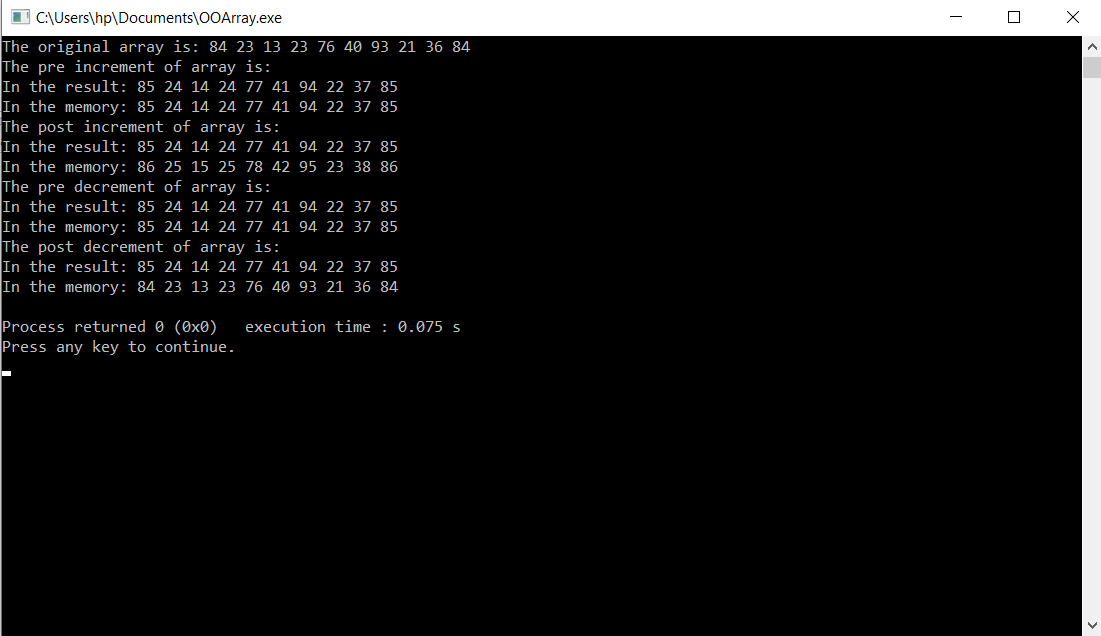
cout<<"In the memory: ";

u1.print();

return 0;

}

**Output:**



**Q12) Write a program to calculate the total mark of a student using the concept of virtual base class.**

**Ans)**

**Code:**

//Student Result.cpp

#include <iostream>

#include <string>

using namespace std;

class Student

{

protected:

string ID;

string name;

public:

void get()

{

cout<<"Enter the student ID: ";

cin>>ID;

cout<<"Enter the student name: ";

cin>>name;

}

};

class Theory\_Marks: public virtual Student

{

protected:

float english;

float french;

float maths;

public:

void getdata()

{

cout<<"Enter the marks of following subjects:"<<endl;

cout<<"English: ";

cin>>english;

cout<<"French: ";

cin>>french;

cout<<"Mathematics: ";

cin>>maths;

}

};

class Sport\_Marks: public virtual Student

{

protected:

float sports\_marks;

public:

void takes()

{

cout<<"Sports Marks: ";

cin>>sports\_marks;

}

};

class Result: public Theory\_Marks, public Sport\_Marks

{

protected:

float total\_marks;

float Marks()

{

total\_marks=Theory\_Marks::english+Theory\_Marks::french+Theory\_Marks::maths+Sport\_Marks::sports\_marks;

return total\_marks;

}

public:

void putdata()

{

cout<<"ID: "<<Student::ID<<endl;

cout<<"Name: "<<Student::name<<endl;

cout<<"Marks obtained in: "<<endl;

cout<<"English: "<<Theory\_Marks::english<<endl;

cout<<"French: "<<Theory\_Marks::french<<endl;

cout<<"Mathematics: "<<Theory\_Marks::maths<<endl;

cout<<"Sports Marks: "<<Sport\_Marks::sports\_marks<<endl;

cout<<"Total marks obtained in all the subjects: "<<Marks()<<endl;

}

};

int main()

{

int n;

Result r[100];

cout<<"Enter the no of students whose record do you want to enter: ";

cin>>n;

if(n!=0)

{

cout<<"Enter the student details and their marks obtained in the exams:"<<endl;

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

{

r[i].get();

r[i].getdata();

r[i].takes();

}

cout<<"The Result is:"<<endl;

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

{

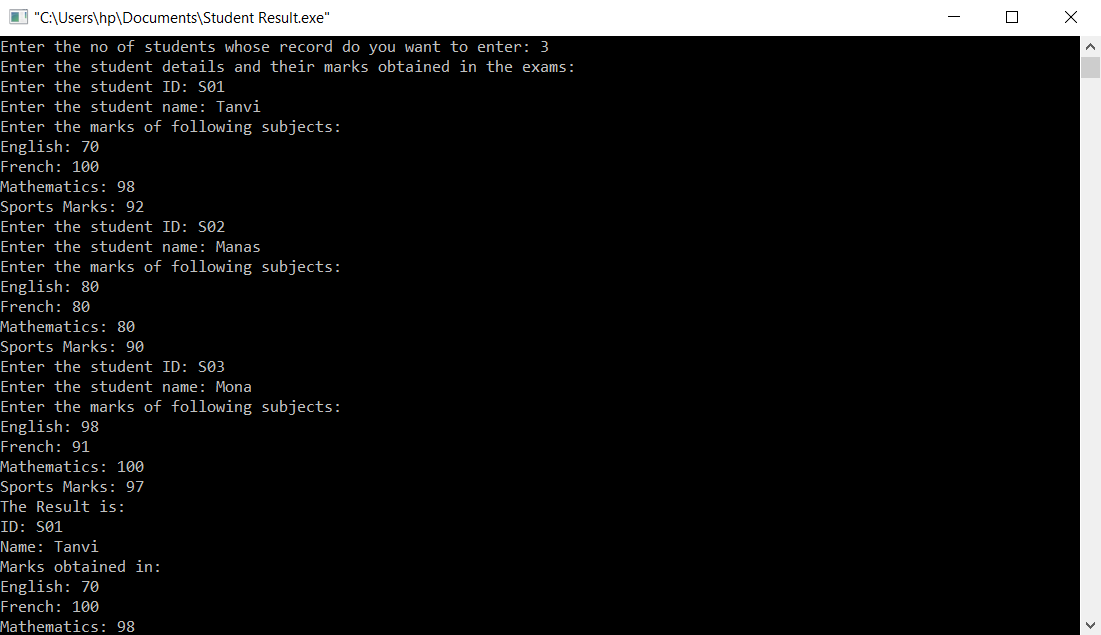
r[i].putdata();

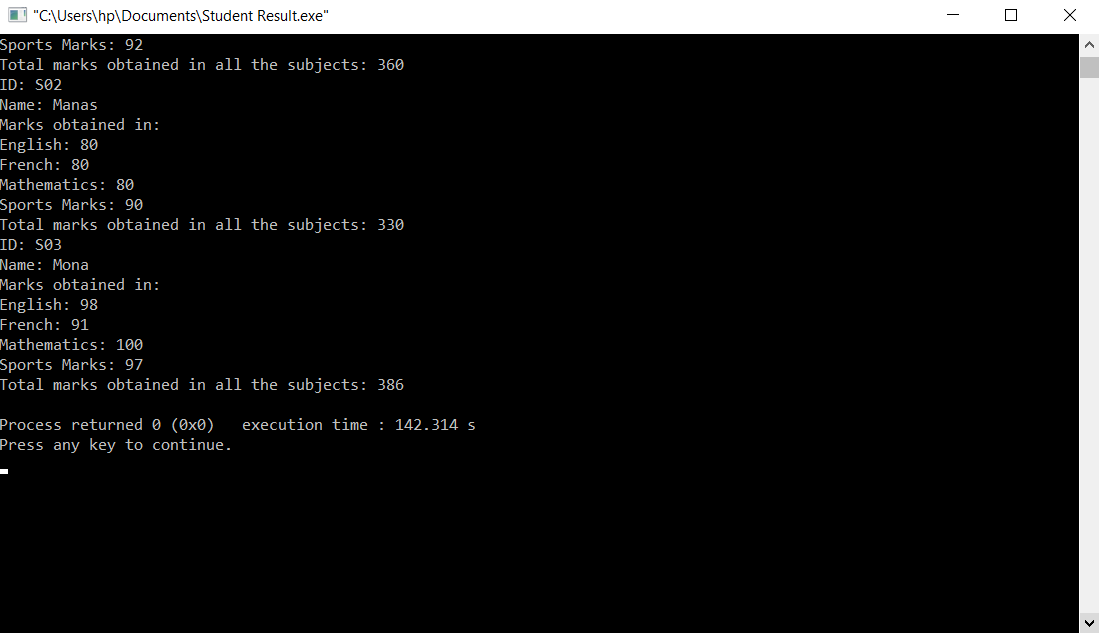
}

}

}

**Output:**





**Q13) Write a C++ program that to perform various operations on strings handling using string class.**

**Ans)**

**Code:**

//String Class.cpp

#include<iostream>

#include<string>

using namespace std;

int main()

{

//Initialization in a string class.

string s1="Physics";

string s2("Mathematics");

string s3;

char \*s4=new char[100];

//Taking input from string class object.

cout<<"Enter the string: ";

getline(cin,s3);

//Taking a character array input.

cout<<"Enter the character array: ";

cin.getline(s4,100);

//Output of string class object.

cout<<"Output is :"<<endl;

cout<<"s1= "<<s1<<endl;

cout<<"s2= "<<s2<<endl;

cout<<"s3= "<<s3<<endl;

//Output of character array.

cout<<"s4= "<<s4<<endl;

//Two swap the two string class objects.

s1.swap(s2);

//Don't do this.--> s3.swap(s4)-->To swap character array with the string class object.

cout<<"After Swapping:"<<endl;

cout<<"s1= "<<s1<<endl;

cout<<"s2= "<<s2<<endl;

//Finding String Objects.

//To find the position of the particular string in the given string order wise.

string s5="Computer System Architecture book is written by M.Morris Mano.";

int pos1=s5.find("Mano");

cout<<"pos1= "<<pos1<<endl;

//To find the position of the particular string that will scan the given string backwards.

int pos2=s5.rfind("Mano");

cout<<"pos2= "<<pos2<<endl;

//To find the position of a string that will match any of the character of the string in the argument from the given string.

int pos3=s5.find\_first\_of("Mano");

cout<<"pos3= "<<pos3<<endl;

//To find the position of a string that will match any of the character of the string in the argument from the given string from backwards.

int pos4=s5.find\_last\_of("Mano");

cout<<"pos4= "<<pos4<<endl;

//To find the position of a string that will not match any of the character of the string in the argument from the given string.

int pos5=s5.find\_first\_not\_of("Mano");

cout<<"pos5= "<<pos5<<endl;

//To find the position of a string that will not match any of the character of the string in the argument from the given string from backwards.

int pos6=s5.find\_last\_not\_of("Mano");

cout<<"pos6= "<<pos6<<endl;

//Modifying String Objects.

string s6("K-means Clustering for Machine Learning");

cout<<"The original string is:"<<endl;

cout<<"s6= "<<s6<<endl;

string s7("Association Rule Mining ");

string s8("Algorithms");

//To erase a particular string from a given string.

s6.erase(0,8);

cout<<"s6= "<<s6<<endl;

//To replace a particular string from another string in a given string.

s6.replace(0,10,s8);

cout<<"s6= "<<s6<<endl;

//To insert a string in a given string

s6.insert(0,s7);

cout<<"s6= "<<s6<<endl;

//To append a string or character in a given string.

s6.append(1,'.');

cout<<"s6= "<<s6<<endl;

//Comparing String Objects

//To compare a particular string from another string.

int a=s1.compare(s2);

if(a==0)

{

cout<<s1<<" matches with "<<s2<<"."<<endl;

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

}

else if(a<0)

{

cout<<s1<<" comes before "<<s2<<"."<<endl;

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

}

else

{

cout<<s2<<" comes before "<<s1<<"."<<endl;

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

}

//Other String Functions

string s9="Geeks for Geeks";

string s10;

cout<<"The original string is:"<<endl;

cout<<"s9= "<<s9<<endl;

//To find the length of a string.

//Method 1

int b=s9.length();

cout<<"The length of the string is: "<<b<<endl;

//Method 2

int c=s9.size();

cout<<"The length of the string is: "<<c<<endl;

//To find out the capacity of the string.

int d=s9.capacity();

cout<<"The capacity of the string is: "<<d<<endl;

//To find out the maximum size of the string.

int e=s9.max\_size();

cout<<"The maximum size of the string is: "<<e<<endl;

//To check whether a string is empty or not.

int f=s10.empty();

if(f)

{

cout<<"The string is empty."<<endl;

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

}

else

{

cout<<"The string is not empty."<<endl;

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

}

//To access a string characters.

//Method 1

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

{

cout<<s9[i];

}

cout<<endl;

//Method 2

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

{

cout<<s9.at(i);

}

cout<<endl;

//To find out the substring of a string.

s10=s9.substr(6,3);

cout<<"s10= "<<s10<<endl;

//Operations On String Class Objects Using Operator Overloading

string s11="Data Structures";

string s12;

string s13="Using C++";

string s14;

string s15;

//To copy the contents of one string into another.

s12=s11;

cout<<"The original string is:"<<endl;

cout<<"s11= "<<s11<<endl;

cout<<"The copied string is:"<<endl;

cout<<"s12= "<<s12<<endl;

//To concatenate two strings.

//Method 1

s14=s11+s13;

cout<<"s14= "<<s14<<endl;

//Method 2

s15=s11+" And Algorithms "+s13+" is written by Dr. ABC";

cout<<"s15= "<<s15<<endl;

//To append one string after an another string.

string s16=" XYZ.";

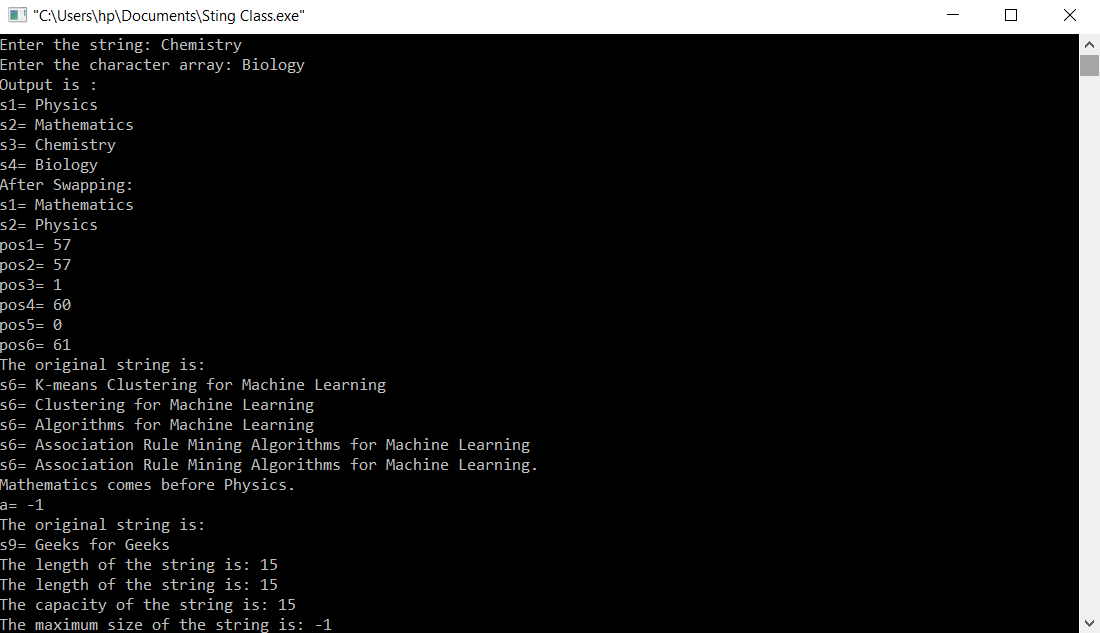
s15+=s16;

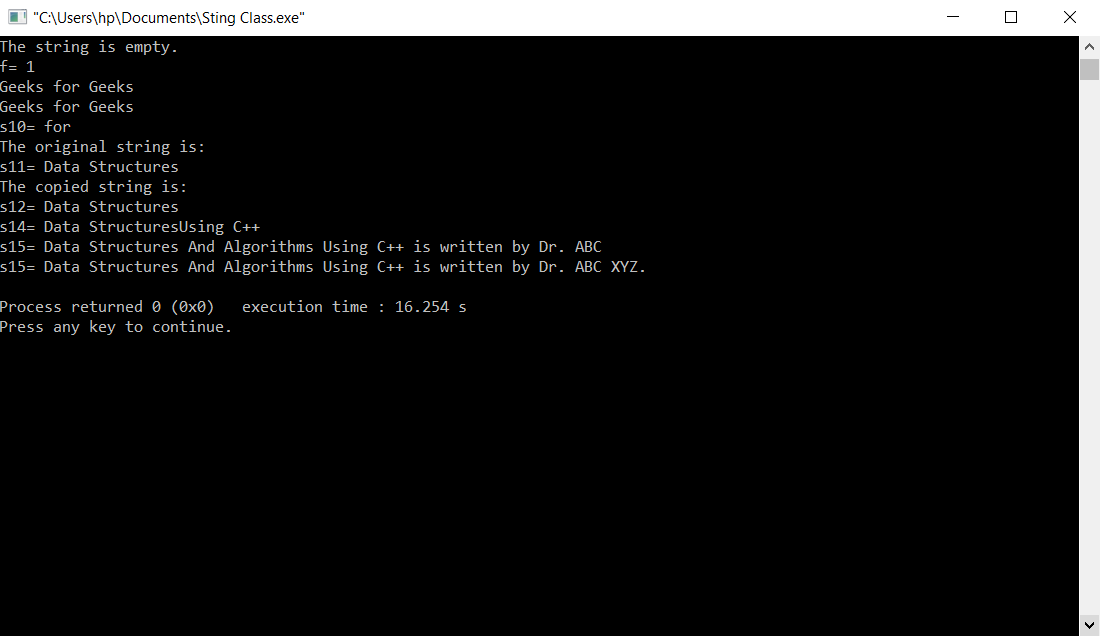
cout<<"s15= "<<s15<<endl;

return 0;

}

**Output:**





**Q14) Write a C++ program to display the contents of a text file.**

**Ans)**

**Code:**

//RWF.cpp  
#include<iostream>

#include<string>

#include<fstream>

using namespace std;

int main()

{

string s;

//To write the contents in a file.

fstream f;

f.open("LabFile.txt",ios::out);

cout<<"Enter the content inside the file: ";

getline(cin,s);

f<<s;

f.close();

//To read the contents from the file.

fstream f1;

f1.open("Labfile.txt",ios::in);

cout<<"The output is: ";

for(int i=0;i<s.size();i++)

{

f1.put(s[i]);

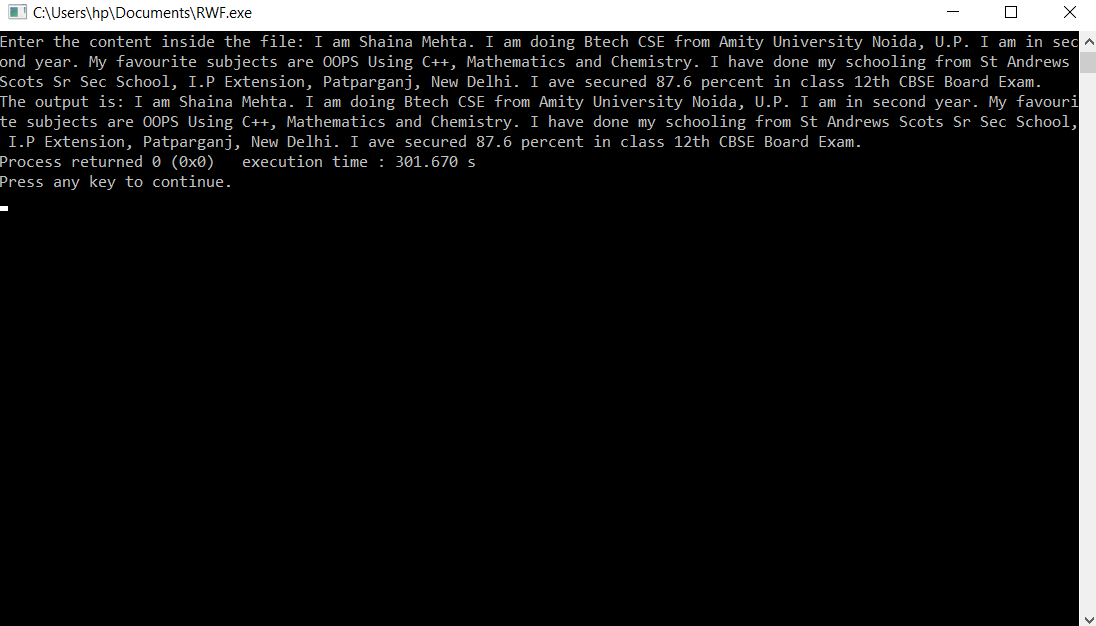
}

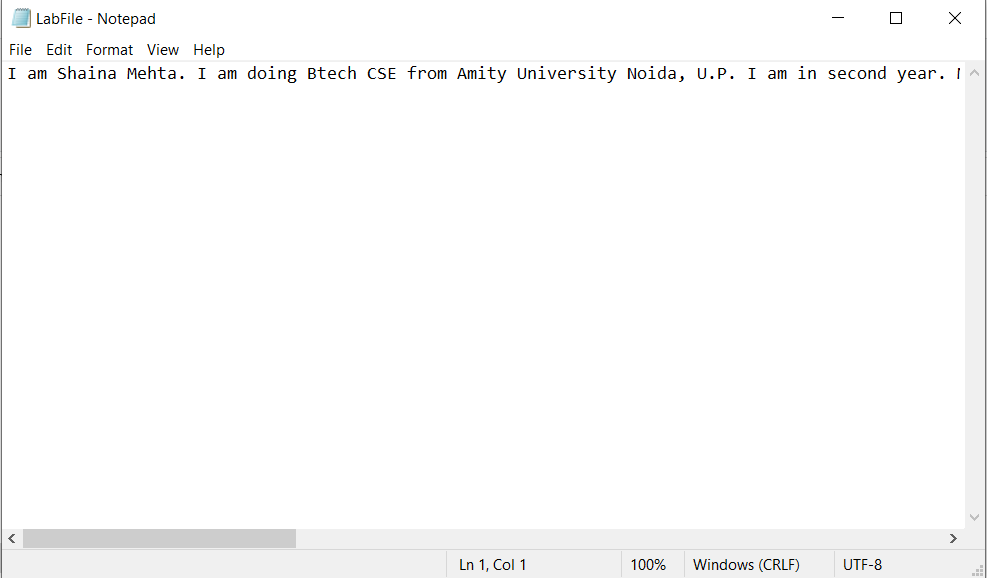
cout<<s;

f1.close();

}

**Output:**





**Q15) Write a program to implement stack functions using templates.**

**Ans)**

**Code:**

//StackUse.cpp

#include <cstddef>

template <typename T>

class Node {

public:

T data;

Node <T> \*next;

Node(T data) {

this->data = data;

next = NULL;

}

};

template <typename T>

class Stack {

// Define the data members

Node <T> \*head;

int size;

public:

Stack() {

// Implement the Constructor

head=NULL;

size=0;

}

/\*----------------- Public Functions of Stack -----------------\*/

int getSize() {

// Implement the getSize() function

return size;

}

bool isEmpty() {

// Implement the isEmpty() function

return size==0;

}

void push(T element) {

// Implement the push() function

Node <T> \*newnode=new Node<T>(element);

if(head==NULL)

{

head=newnode;

}

else

{

newnode->next=head;

head=newnode;

}

size++;

}

T pop() {

// Implement the pop() function

if(head==NULL)

{

return -1;

}

else

{

size--;

Node <T> \*a=head;

int b=a->data;

head=head->next;

delete a;

return b;

}

}

T top() {

// Implement the top() function

if(head==NULL)

{

return -1;

}

else

{

return head->data;

}

}

};

//Stackmain.cpp

#include <iostream>

using namespace std;

#include "StackUse.cpp"

int main() {

Stack <int> st;

int q;

cin >> q;

while (q--) {

cout<<"Main Menu"<<endl;

cout<<"1.To push the element into the stack."<<endl;

cout<<"2.To Pop the element out of the stack."<<endl;

cout<<"3.To show the topmost element of the stack."<<endl;

cout<<"4.Calculation of the size."<<endl;

cout<<"5.To Check whether the stack is empty or not."<<endl;

int choice, input;

cout<<"Enter the choice:"<<endl;

cin >> choice;

switch (choice) {

case 1:

cout<<"Enter the element in the stack:"<<endl;

cin >> input;

st.push(input);

break;

case 2:

cout <<"The element deleted from the stack is:"<< st.pop() << "\n";

break;

case 3:

cout <<"The element present on the top of the stack is:"<< st.top() << "\n";

break;

case 4:

cout <<"The size of the stack is:"<< st.getSize() << "\n";

break;

case 5:

cout <<"The stack is:"<< ((st.isEmpty()) ? "true\n" : "false\n");

break;

default:

cout<<"Invalid Entry."<<endl;

}

}

return 0;

}

**Output:**

