

**Program-1**

#include<iostream>

using namespace std;

class superr

{

public:

superr();

~superr();

};

superr::superr()

{

cout<<"Class superr Constructor"<<endl;

}

superr::~superr()

{

cout<<"Class superr Destructor"<<endl;

}

class B:public superr

{

public:

B();

~B();

};

B::B()

{

cout<<"Class B Constructor"<<endl;

}

B::~B()

{

cout<<"Class B Destructor"<<endl;

}

class A:public B

{

public:

A();

~A();

};

A::A()

{

cout<<"Class A Constructor"<<endl;

}

A::~A()

{

cout<<"Class A Destructor"<<endl;

}

int main()

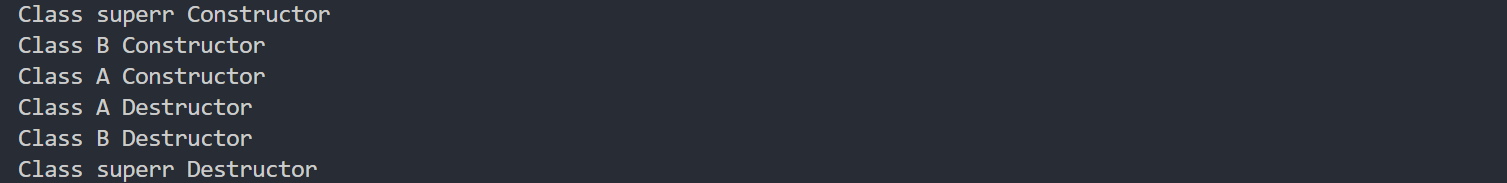
{

A a;

return 0;

}

**Output**:-



**Program-2**

#include<iostream>

using namespace std;

class AABC

{

private:

public:

AABC();

~AABC();

};

AABC::AABC()

{

cout<<"Inside Virtual Class Constructor- [AABC]"<<endl;

}

AABC::~AABC()

{

cout<<"Inside Virtual Class Destructor- [AABC]"<<endl;

}

class ABC

{

public:

ABC();

~ABC();

};

ABC::ABC()

{

cout<<"Child Class One Constructor- [ABC]"<<endl;

}

ABC::~ABC()

{

cout<<"Child Class One Destructor- [ABC]"<<endl;

}

class child1:public ABC,public virtual AABC

{

public:

child1();

~child1();

};

child1::child1()

{

cout<<"Inside Child Class Constructor- [child1]"<<endl;

}

child1::~child1()

{

cout<<"Inside Child Class Destructor- [child1]"<<endl;

}

int main()

{

child1 c1;

return 0;

}

**Program-3**

#include<iostream>

using namespace std;

class Student{

public:

string name;

long roll;

int age;

public:

Student(string name,long roll,int age){

this->name=name;

this->roll=roll;

this->age=age;

}

};

class test : public Student

{

public:

int \*arr=new int[5];

int sum=0;

test(string name,long roll,int age):Student(name,roll,age){

cout<<"Enter Marks \n";

sum=0;

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

{

cin>>arr[i];

sum=sum+arr[i];

}

}

void totalMarks(){

cout<<"Total Marks obtained in Academics is "<<sum<<endl;

}

};

class Sports

{

public:

int SportScore;

Sports(int SportScore){

this->SportScore=SportScore;

}

};

class Result:public Sports,public test

{

public:

Result(string name,long roll,int age,int score):Sports(score),test(name,roll,age){}

void DisplayDetails(){

int total=Sports::SportScore+test::sum;

float percent=(total/600.0)\*100.0;

cout<<"Details of the Student\n";

cout<<"\nName:"<<Student::name<<"\nStudent Roll"<<Student::roll<<"\nStudent Age"<<Student::age<<"\nSports Score "<<Sports::SportScore<<endl;

cout<<"Total Marks "<<total<<"\nPercentage Secured is "<<percent<<endl;

}

};

int main()

{

Result r1("AURO",22057020,40,58);

r1.DisplayDetails();

return 0;

}

