11.WriteaC++programtocreatemultilevelinheritance.(Hint:ClassesA1,A2,A3)

Program:

#include<iostream> usingnamespacestd; classA1{ public: inta; A1(){ cin>>a;

}

voiddisplay1(){ cout<<"ais"<<a<<endl;

}

};

classA2:publicA1{ public: intb; A2(){ cin>>b;

}

voiddisplay2(){ cout<<"bis"<<b<<endl;

}

};

classA3:publicA2{ public: intc; A3(){ cin>>c;

}

voiddisplay3(){ cout<<"cis"<<c<<endl;

}

};

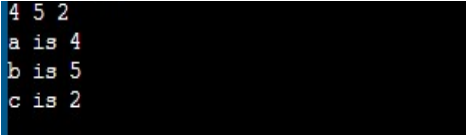
intmain(){ A3obj;

obj.display1(); obj.display2(); obj.display3();

return0;

}

Output:



12.(A):WriteaC++Programtoillustrateearlybinding.

Program:

#include<iostream> usingnamespacestd;

classAdd

{

public:

intadd(inta,intb){ returna+b;

}

intadd(inta,intb,intc){ returna+b+c;

}

};

intmain()

{

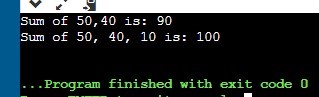
Adda;

cout<<"Sumof50,40is:"<<a.add(50,40)<<endl; cout<<"Sumof50,40,10is:"<<a.add(50,40,10)<<endl;

return0;

}

Output:



12.(b):WriteaC++Programtoillustratelatebinding.

Program:

#include<iostream> usingnamespacestd;

classBase

{

public:

virtualvoidshow(){

cout<<"InBase\n";

}

};

classDerived:publicBase

{

public:

voidshow(){

cout<<"InDerived\n";

}

};

intmain(void)

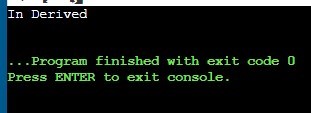
{

Base\*bp=newDerived; bp->show();//RUN-TIMEPOLYMORPHISM

return0;

}

Output:



12(c):WriteaC++ProgramtoillustrateconceptofVirtualFunctions.

Program:

#include<iostream> usingnamespacestd; classA{ intx=10; public:

virtualvoiddisplay(){ cout<<x;

}

};

classB:publicA{ inty=20; public:

voiddisplay(){ cout<<"Valueofyis:"<<y;

}

};

intmain(){

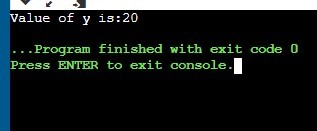
A\*a; Bb; a=&b;

a->display();

return0;

}

Output:



12(d):WriteaC++ProgramonAbstractclasses.

Program:

#include<iostream> usingnamespacestd;

classbase

{

inta; public:

virtualvoidfunction()=0; intgetX(){ returna;

}

};

classderived:publicbase

{

intb; public:

voidfunction(){

cout<<"functioniscalled";

}

};

intmain()

{

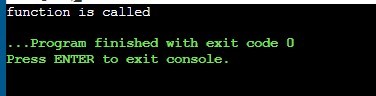
derivedd1;

d1.function();

return0;

}

Output:



13(a):WriteaC++Programfordividebyzerousingtry&catchblocks

Program:

#include<iostream> usingnamespacestd;

intmain(){

intx,y; cout<<"numberatoris"; cin>>x;

try{

cout<<"Denominatoris"; cin>>y; if(y<=0){

throwy;

}

cout<<"Answeris"<<x/y<<endl;

}

catch(inta){

cout<<"Exceptioncaught!!"<<endl;

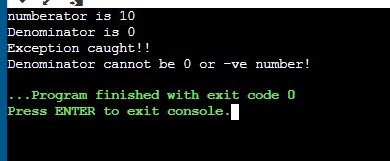
cout<<"Denominatorcannotbe0or-venumber!";

}

return0;

}

Output:



13(b):WriteaC++Programfordividebyzerousingmultiplecatch.

Program:

#include<iostream> #include<exception> usingnamespacestd; intmain(){ inta,b; floatc;

cout<<"Enterthenumeratoranddenominator:";

cin>>a>>b; try{

if(b==0){

throw(b);

}

c=a/b;

cout<<"Resultis:"<<c<<endl;

}

catch(inti){ cout<<"exceptioniscatch"<<endl;

}

catch(...){

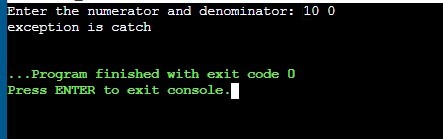
cout<<"denominatorcannotbezero!!"<<endl;

}

return0;

}

Output:



14(a):WriteaC++toillustratetheconceptsofconsoleI/Ooperations.

Program:

#include<iostream> usingnamespacestd; intmain()

{

charx;

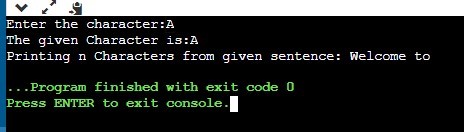
//usedtoscanasinglechar cout<<"Enterthecharacter:"; cin.get(x);

//usedtoputasinglecharontothescreen. cout<<"ThegivenCharacteris:"; cout.put(x);

//ncountcharacterfromarray cout.write("\nWelcometoHyderabad",10);

}

Output:



14(b):WriteaC++onfilestreams.

Program:

//WriteaC++onfilestreams

#include<fstream> #include<iostream> usingnamespacestd;

intmain()

{

chardata[100];

ofstreamoutfile; outfile.open("afile.dat");

cout<<"Writingtothefile"<<endl; cout<<"Enteryourname:";

cin.getline(data,100); outfile<<data<<endl; cout<<"Enteryourrollnumber:"; cin>>data; cin.ignore(); outfile<<data<<endl; outfile.close(); ifstreaminfile;

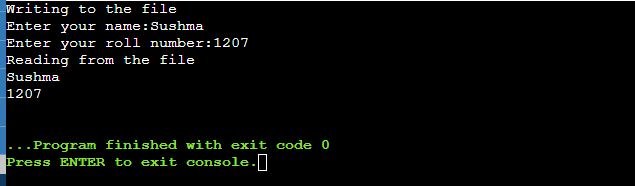
infile.open("afile.dat");

cout<<"Readingfromthefile"<<endl; infile>>data; cout<<data<<endl; infile>>data; cout<<data<<endl;

infile.close();

return0;

}

Output: 

15(a):WriteaC++programtooverloadunaryoperator.

Program:

#include<iostream> usingnamespacestd; classsample{

intnum; public:

sample(){ cout<<"Previouscount:";

cin>>num;

}

voidoperator++(){ num=num+2;

}

voidPrint(){ cout<<"Countis:"<<num;

}

};

intmain()

{

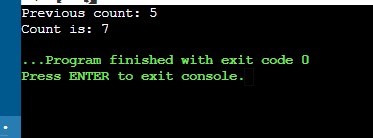
samples1;

++s1;//callingofafunction"voidoperator++()" s1.Print();

return0;

}

Output:



15(b):WriteaC++programtooverloadbinaryoperator.

Program:

#include<iostream> usingnamespacestd;

classA

{

intx; public: A(){}

A(inti)

{

x=i;

}

voidoperator+(A);

voiddisplay();

};

voidA::operator+(Aa)

{

intm=x+a.x;

cout<<"Theresultoftheadditionoftwoobjectsis:"<<m;

}

intmain()

{

Aa1(5);

Aa2(4); a1+a2;

return0;

}

Output:

