1. Program to calculate sum of two numbers using class.

#include<iostream>

using namespace std;

class add

{

public:

int a,b,c;

void sum();

};

void add::sum()

{

cout<<"enter the numbers";

cin>>a>>b;

c=a+b;

cout<<"result is "<<c;

}

main()

{

add ob;

ob.sum();

}

output

enter the numbers

3

4

result is 7

2. Program to add points using class.

#include<iostream>

using namespace std;

class point

{

int x,y;

public:

void read();

void display();

point add(point);

}p1,p2,p3;

void point ::read()

{

cin>>x>>y;

}

void point::display()

{

cout<<x<<" "<<y<<"\n";

}

point point:: add(point p1)

{

p3.x=x+p1.x;

p3.y=y+p1.y;

return p3;

}

main()

{

cout<<"Enter first point\n";

p1.read();

cout<<"Enter second point\n";

p2.read();

cout<<"First and second point is\n";

p1.display();

p2.display();

cout<<"sum is\n";

p3=p2.add(p1);

p3.y=y+p1.y;

return p3;

}

main()

{

cout<<"Enter first point\n";

p1.read();

cout<<"Enter second point\n";

p2.read();

cout<<"First and second point is\n";

p1.display();

p2.display();

cout<<"sum is\n";

p3=p2.add(p1);

p3.display();

}

output

Enter first point

2

3

Enter second point

4

5

First and second point is

2 3

4 5

sum is

6 8

3. Program to add two complex numbers using class.

#include<iostream>

using namespace std;

class complx

{

public:

int real1,real2,img1,img2;

void read()

{

cout<<"enter real and imgnry part of first num";

cin>>real1>>img1;

cout<<"enter real and img part of 2nd num";

cin>>real2>>img2;

}

void add()

{

int r=real1+real2;

int im=img1+img2;

cout<<"sum="<<r<<"+"<<im<<"i";

}

};

main()

{

complx c;

c.read();

c.add();

}

output

enter real and imgnry part of first num

6

8

enter real and img part of 2nd num5

3

sum=11+11i

4. Write a program to create a class String with member variable a character array, define constructor, destructor and member functions to copying a string to another and concatenating two strings.

#include<iostream>

#include<string.h>

using namespace std;

class strng

{

public:

char a[20],b[30];

strng()

{

cout<<"Enter a string\n";

cin>>a;

cout<<"Enter another string\n";

cin>>b;

}

void copy()

{

strcpy(b,a);

cout<<"\ncopy of a="<<b;

}

void add()

{

cout<<"\nconcatinated string is\n"<<strcat(a,b);

}

~strng()

{

cout<<"\ndestroy string";

}

};

main()

{

strng s;

s.add();

s.copy();

}

output

Enter a string

good

Enter another string

morning

concatinated string is

goodmorning

copy of a=goodmorning

5. Write a program to create class Distance with member variables feet and inch. Define constructor destructor and a member function to add two distances using constructors.

#include<iostream>

using namespace std;

class distnc

{

public:

int feet,inch;

distnc()

{

feet=10;

inch=5;

}

distnc(int a,int b)

{

feet=a;

inch=b;

}

void disp()

{

cout<<"distance="<<feet<<"."<<inch<<"\n";

}

};

main()

{

distnc d1,d2(20,30),d3;

d1.disp();

d2.disp();

d3.feet=d1.feet+d2.feet;

d3.inch=d1.inch+d2.inch;

d3.disp();

}

output

distance=10.5

distance=20.30

distance=30.35

6. Write a C++ program to find the largest of three numbers using inline function.

#include<iostream>

using namespace std;

inline void large(int a,int b,int c)

{

if(a>b)

{

if(a>c)

{

cout<<a<<" is largest\n";

}

else

{

cout<<c<<" is largest\n";

}

}

else

{

if(b>c)

{

cout<<b<<" is largest\n";

}

else

{

cout<<c<<" is largest\n";

}

}

}

main()

{

int a,b,c;

cout<<"Enter three numbers\n";

cin>>a>>b>>c;

large(a,b,c);

}

output

Enter three numbers

3

4

45

45 is largest

7. Write a program to add two member variables of a class using friend function.

#include<iostream>

using namespace std;

class myclass

{

int a,b;

public:

friend int sum(myclass x);

void set\_ab(int i,int j);

};

void myclass::set\_ab(int i,int j)

{

a=i;

b=j;

}

int sum(myclass x)

{

return x.a+x.b;

}

main()

{

myclass n;

n.set\_ab(3,5);

cout<<sum(n);

}

output

8

8. Write a program to add two complex number using friend function.

#include<iostream>

usingnamespace std;

class complex

{

int a,b;

public:

void read()

{

cout<<"Enter comlex number:";

cin>>a>>b;

}

void display()

{

cout<<a<<"+"<<b<<"i"<<"\n";

}

friend complex sum(complex,complex);

};

complex sum(complex ob1,complex ob2)

{

complex temp;

temp.a=ob1.a+ob2.a;

temp.b=ob1.b+ob2.b;

return temp;

}

main()

{

complex ob1,ob2,ob3;

ob1.read();

ob2.read();

cout<<"Two complex numbers are:\n";

ob1.display();

ob2.display();

cout<<"Sum=";

ob3=sum(ob1,ob2);

ob3.display();

}

output

Enter comlex number:4

5

Enter comlex number:6

7

Two complex numbers are:

4+5i

6+7i

Sum=10+12i

9.Write a program to find minimum of two values for demonstrate friend classes.

#include<iostream>

using namespace std;

class fclass

{

public:

int a,b;

fclass(int i,int j)

{

a=i;

b=j;

}

friend class Min;

};

class Min

{

public:

int min(fclass x);

};

int Min::min(fclass x)

{

return x.a<x.b?x.a:x.b;

}

main()

{

fclass ob(10,20);

Min m;

cout<<m.min(ob);

}

output

10

10. Write a program to find volume of cube , cylinder and rectangle using function overloading.

#include<iostream>

using namespace std;

class overload

{

public:

int volume(int);

float volume(float,float);

float volume(float,float,float);

};

int overload::volume(int a)

{

return(a\*a\*a);

}

float overload::volume(float r,float h)

{

return(3,14\*r\*r\*h);

}

float overload::volume(float l,float b,float h2)

{

return(l\*b\*h2);

}

main()

{

int a,l,b,h,r,h2;

overload ob;

cout<<"Enter side of cube";

cin>>a;

cout<<" \nVolume of a cube is ="<<ob.volume(a);

cout<<" \nEnter radious and height of cylinder";

cin>>r>>h;

cout<<" \nvolume of cylender is ="<<ob.volume(r,h)<<"\n";

cout<<"Enter length,breadth and height of rectangle";

cin>>l>>b>>h2;

cout<<" \nVolume of rectangle ="<<ob.volume(l,b,h2)<<"\n";

}

output

enter side of cube

8

volume of cube is

512enter radius and height of the cylinder

4

6

volume of cylinder is=

301.44

enter length,bredth and height of thr rectangle

4

5

6

volume of rectangle=120

11.Create a 'MATRIX' class of size m X n. Overload the â€˜+â€™ and â€˜\*â€™ operator to add and multiply two MATRIX objects.

#include<iostream>

using namespace std;

class matrix

{

public:

int a[50][50],i,j,k;

int r,c;

void read\_size()

{

cout<<"Enter row size and column size:";

cin>>r>>c;

}

void read()

{

cout<<"Enter values:";

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

{

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

{

cin>>a[i][j];

}

}

}

void display()

{

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

{

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

{

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

}

cout<<"\n";

}

matrix operator +(matrix);

matrix operator \*(matrix);

};

matrix matrix::operator +(matrix ob)

{

matrix temp;

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

{

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

{

temp.a[i][j]=0;

temp.a[i][j]=a[i][j]+ob.a[i][j];

}

temp.r=r;

temp.c=c;

}

return (temp);

}

matrix matrix::operator \*(matrix m)

{

matrix temp2;

temp2.r=r;

temp2.c=m.c;

for(i=0;i<temp2.r;i++)

{

for(j=0;j<temp2.c;j++)

{

temp2.a[i][j]=0;

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

{

temp2.a[i][j]=temp2.a[i][j]+(a[i][k]\*m.a[k][j]);

}

}

}

return temp2;

}

main()

{

matrix m1,m2,m3,m4;

m1.read\_size();

m2.read\_size();

m1.read();

m2.read();

cout<<"First matrix is:\n";

m1.display();

cout<<"Second matrix is:\n";

m2.display();

if((m1.r==m2.r)&&(m1.c==m2.c))

{

m3=m1+m2;

cout<<"SUM IS:\n";

m3.display();

}

else

{

cout<<"Addition not possible\n";

}

if(m1.r==m2.c)

{

m4=m1\*m2;

cout<<"PRODUCT OF TWO MATRIX IS:\n";

m4.display();

}

else

{

cout<<"Multiplication Not possible\n";

}

}

output

Enter row size and column size:3

3

Enter row size and column size:3

3

Enter values:1

2

3

4

5

6

7

6

5

Enter values:4

3

2

3

4

5

6

7

7

First matrix is:

1 2 3

4 5 6

7 6 5

Second matrix is:

4 3 2

3 4 5

6 7 7

SUM IS:

5 5 5

7 9 11

13 13 12

PRODUCT OF TWO MATRIX IS:

28 32 33

67 74 75

76 80 79

12.Arithmetic Operations on Complex Number using Operator Overloading

#include<iostream>

#include<string>

using namespace std;

class complex

{

int i,r;

public:

void read()

{

cout<<"\nEnter Real Part:";

cin>>r;

cout<<"Enter Imaginary Part:";

cin>>i;

}

void display()

{

cout<<"\n= "<<r<<"+"<<i<<"i";

}

complex operator+(complex a2)

{

complex a;

a.r=r+a2.r;

a.i=i+a2.i;

return a;

}

complex operator-(complex a2)

{

complex a;

a.r=r-a2.r;

a.i=i-a2.i;

return a;

}

complex operator\*(complex a2)

{

complex a;

a.r=(r\*a2.r)-(i\*a2.i);

a.i=(r\*a2.i)+(i\*a2.r);

return a;

}

complex operator/(complex a2)

{

complex a;

a.r=((r\*a2.r)+(i\*a2.i))/((a2.r\*a2.r)+(a2.i\*a2.i));

a.i=((i\*a2.r)-(r\*a2.i))/((a2.r\*a2.r)+(a2.i\*a2.i));

return a;

}

};

main()

{

int ch;

complex a,b,c;

do

{

cout<<"\n1.Addition 2.Substraction";

cout<<" 3.Mulitplication 4.Division 5.Exit\n";

cout<<"\nEnter the choice :";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\nEnter The First Complex Number:";

a.read();

a.display();

cout<<"\nEnter The Second Complex Number:";

b.read();

b.display();

c=a+b;

c.display();

break;

case 2:

cout<<"\nEnter The First Complex Number:";

a.read();

a.display();

cout<<"\nEnter The Second Complex Number:";

b.read();

b.display();

c=b-a;

c.display();

break;

case 3:

cout<<"\nEnter The First Complex Number:";

a.read();

a.display();

cout<<"\nEnter The Second Complex Number:";

b.read();

b.display();

c=a\*b;

c.display();

break;

case 4:

cout<<"\nEnter The First Complex Number:";

a.read();

a.display();

cout<<"\nEnter The Second Complex Number:";

b.read();

b.display();

c=a/b;

c.display();

break;

}

}while(ch!=5);

}

output

1.Addition 2.Substraction 3.Mulitplication 4.Division 5.Exit

Enter the choice :1

Enter The First Complex Number:

Enter Real Part:2

Enter Imaginary Part:3

= 2+3i

Enter The Second Complex Number:

Enter Real Part:4

Enter Imaginary Part:5

= 4+5i

= 6+8i

1.Addition 2.Substraction 3.Mulitplication 4.Division 5.Exit

Enter the choice :2

Enter The First Complex Number:

Enter Real Part:3

Enter Imaginary Part:6

= 3+6i

Enter The Second Complex Number:

Enter Real Part:5

Enter Imaginary Part:3

= 5+3i

= 2+-3i

1.Addition 2.Substraction 3.Mulitplication 4.Division 5.Exit

Enter the choice :3

Enter The First Complex Number:

Enter Real Part:2

Enter Imaginary Part:3

= 2+3i

Enter The Second Complex Number:

Enter Real Part:1

Enter Imaginary Part:4

= 1+4i

= -10+11i

1.Addition 2.Substraction 3.Mulitplication 4.Division 5.Exit

Enter the choice :4

Enter The First Complex Number:

Enter Real Part:6

Enter Imaginary Part:2

= 6+2i

Enter The Second Complex Number:

Enter Real Part:4

Enter Imaginary Part:8

= 4+8i

= 0+0i

1.Addition 2.Substraction 3.Mulitplication 4.Division 5.Exit

Enter the choice :5

13. Write a program to implement inheritance.

#include<iostream>

using namespace std;

class emp

{

public:

int eno;

char name[10],des[20];

void get()

{

cout<<"Enter the employee number";

cin>>eno;

cout<<"Enter the employee name";

cin>>name;

cout<<"Enter the designation";

cin>>des;

}

};

class salary:public emp

{

float bp,hra,da,pf,np;

public:

void get1()

{

cout<<"Enter the basic pay";

cin>>bp;

cout<<"Enter the humen resource Allowance";

cin>>da;

cout<<"Enter the profitability Fund";

cin>>pf;

}

void calculate()

{

np=bp+hra+da-pf;

}

void display()

{

cout<<eno<<"\t"<<name<<"\t"<<des<<"\t"<<hra<<"\t"<<da<<"\t"<<pf<<"\t"<<np<<"\n";

}

};

main()

{

int i,n;

char ch;

salary s[10];

cout<<"Enter the number of employee";

cin>>n;

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

{

s[i].get();

s[i].get1();

s[i].calculate();

}

cout<<"\ne\_no\te\_name\tdes\tbp\thra\tda\tpf\tnp\n";

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

{

s[i].display();

}

}

output

enter the number of employees

2

Enter the employee number

1

Enter the employee name

nathiya

Enter the destination

chpzha

enter the basic pay

3000

enter the human resource allowance

200

enter the profitability fund

340

Enter the employee number

2

Enter the employee name

chinchu

Enter the destination

alakode

enter the basic pay

2000

enter the human resource allowance

300

enter the profitability fund

120

eno ename des bp hra da pf np

1 nathiya chpzha 3000 8.40779e-45 200 340 2860

2 chinchu alakode 2000 -0.116702 300 120 2179.88

14. Write a program to implement virtual base class.

#include<iostream>

using namespace std;

class base

{

public:

int i;

};

class derived1:virtual public base

{

public:

int j;

};

class derived2:virtual public base

{

public:

int k;

};

class derived3:public derived1,public derived2

{

public:

int sum;

};

main()

{

derived3 ob;

ob.i=10;

ob.j=20;

ob.k=30;

ob.sum=ob.i+ob.j+ob.k;

cout<<ob.i<<" ";

cout<<ob.j<<" "<<ob.k<<" ";

cout<<ob.sum;

return 0;

}

output

10 20 30 60

15. Program to implement polymorphism.

#include<iostream>

using namespace std;

class rectangle

{

float a,b,rect;

public:

void area()

{

cout<<"Enter width and height of rectangle";

cin>>a>>b;

rect=a\*b;

cout<<"Area of rectangle="<<rect<<"\n";

}

};

class square

{

float a,sq;

public:

void area()

{

cout<<"Enter side of square";

cin>>a;

sq=a\*a;

cout<<"Area of square="<<sq<<"\n";

}

};

class circle

{

float r,circ;

public:

void area()

{

cout<<"Enter radious of circle";

cin>>r;

circ=3.14\*r\*r;

cout<<"Area of circle="<<circ<<"\n";

}

};

main()

{

rectangle ob;

square ob1;

circle ob2;

ob.area();

ob1.area();

ob2.area();

}

output

Enter width and height of rectangle

3

4

Area of rectangle=12

Enter side of square

6

Area of square=36

Enter radious of circle

4

Area of circle=50.24

16. Program to implement virtual functions.

#include<iostream>

using namespace std;

class base

{

public:

virtual void vfunc()

{

cout<<"this is base's vfunc()\n";

}

};

class derived1:public base

{

public:

void vfunc()

{

cout<<"This is derived 1's vfunc()\n";

}

};

class derived2:public base

{

public:

void vfunc()

{

cout<<"this is derived 2's vfunc()\n";

}

};

main()

{

base \*p,b;

derived1 d1;

derived2 d2;

p=&b;

p->vfunc();

p=&d1;

p->vfunc();

p=&d2;

p->vfunc();

}

output

this is base's vfunc()

This is derived 1's vfunc()

this is derived 2's vfunc()

17. Write a program to implement C++ files.

#include<iostream>

#include<fstream>

using namespace std;

class student

{

private:

int rollno,m1,m2,m3;

char name[20];

int total;

float avg;

public:

void getdata()

{

cout<<"Enter roll number"<<"\n";

cin>>rollno;

cout<<"Enter student name"<<"\n";

cin>>name;

cout<<"Enter mark of three subjects"<<"\n";

cin>>m1>>m2>>m3;

}

void putdata()

{

cout<<"Roll Bo"<<rollno<<"\n";

cout<<"Name"<<name<<"\n";

cout<<"Marks"<<m1<<" "<<m2<<" "<<m3<<" "<<"\n";

total=m1+m2+m3;

avg=total/3;

cout<<"TotalMark"<<total<<"\n";

cout<<"Average"<<avg<<"\n";

}

};

main()

{

cout<<"C++ FILES \n";

cout<<"\n";

student st;

st.getdata();

ofstream outfile("mark.out");

outfile.write((char\*)&st,sizeof(st));

outfile.close();

cout<<"Display file";

ifstream infile("mark.out");

infile.read((char\*)&st,sizeof(st));

st.putdata();

}

output

C++ FILES

Enter roll number

1

Enter student name

nathiya

Enter mark of three subjects

45

56

65

Display fileRoll Bo1

Namenathiya

Marks45 56 65

TotalMark166

Average55

18.STL

#include<iostream>

#include<vector>

#include<cctype>

using namespace std;

int main()

{

vector<char>v(10);

unsigned int i;

cout<<"Size="<<v.size()<<endl;

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

v[i]=i+'a';

cout<<"Current Elements\n";

for(i=0;i<v.size();i++)

cout<<v[i]<<" ";

cout<<"\n\n";

cout<<"Expanding Vector:\n";

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

v.push\_back(i+10+'a');

cout<<"size new ="<<v.size()<<endl;

cout<<"Current contents\n";

for(i=0;i<v.size();i++)

cout<<v[i]<<" ";

cout<<"\n\n";

for(i=0;i<v.size();i++)

v[i]=toupper(v[i]);

cout<<"Modified Contents\n";

for(i=0;i<v.size();i++)

cout<<v[i]<<" ";

cout<<"\n\n";

}

output

Size=10

Current Elements

a b c d e f g h i j

Expanding Vector:

size new =20

Current contents

a b c d e f g h i j k l m n o p q r s t

Modified Contents

A B C D E F G H I J K L M N O P Q R S T

19.Write a program to demonstrate IO based program.

#include&lt;iostream&gt;

#include&lt;strstream&gt;

usingnamespace std;

main()

{

char iostr[80];

strstream strio(iostr,sizeof(iostr),ios::in|ios::out);

int a,b;

char str[80];

strio&lt;&lt;&quot;60 90 testing&quot;;

strio&gt;&gt;a&gt;&gt;b&gt;&gt;str;

cout&lt;&lt;a&lt;&lt;&quot; &quot;&lt;&lt;b&lt;&lt;&quot; &quot;&lt;&lt;str&lt;&lt;endl;

}

OUTPUT:

60 90 testing

20.Write a program to demonstrate IO based program.

#include&lt;iostream&gt;

#include&lt;cstring&gt;

using namespace std;

class phonebook

{

char name[20];

int areacode,prefix;

longint num;

public:

phonebook()

{

}

phonebook(char\*n,int a,int p,int nm)

{

strcpy(name,n);

areacode=a;

prefix=p;

num=nm;

}

friend ostream &amp;operator&lt;&lt;(ostream &amp;stream,phonebook o);

friend istream &amp;operator&gt;&gt;(istream &amp;stream,phonebook &amp;o);

};

ostream &amp;operator&lt;&lt;(ostream &amp;stream,phonebook o)

{

stream&lt;&lt;o.name&lt;&lt;&quot; &quot;;

stream&lt;&lt;&quot;(&quot;&lt;&lt;o.areacode&lt;&lt;&quot;)&quot;;

stream&lt;&lt;o.prefix&lt;&lt;&quot;-&quot;&lt;&lt;o.num&lt;&lt;&quot;\n&quot;;

return stream;

}

istream &amp;operator&gt;&gt;(istream &amp;stream,phonebook &amp;o)

{

cout&lt;&lt;&quot;Enter name:&quot;;

stream&gt;&gt;o.name;

cout&lt;&lt;&quot;Enter areacode:&quot;;

stream&gt;&gt;o.areacode;

cout&lt;&lt;&quot;Enter prefix:&quot;;

stream&gt;&gt;o.prefix;

cout&lt;&lt;&quot;Enter number:&quot;;

stream&gt;&gt;o.num;

cout&lt;&lt;&quot;\n&quot;;

return stream;

}

main()

{

phonebook ob;

cin&gt;&gt;ob;

cout&lt;&lt;ob;

}

OUTPUT:

Enter name:Nathiya

Enter areacode:456

Enter prefix:91

Enter number:3344556677

Nathiya (456)91-3344556677