/\*

Vinay

Roll No.- 7118

Program-27 : Unary Operator Overloading

\*/

#include<iostream>

using namespace std;

class space{

    int x;

    int y;

    int z;

    public:

    void getdata(int a,int b,int c);

    void display(void);

    void operator-();

};

void space :: getdata(int a,int b,int c){

    x=a;

    y=b;

    z=c;

}

void space :: display(void){

    cout << "x = " << x << " ";

    cout << "Y = " << y << " ";

    cout << "Z =" << z << " ";

}

void space :: operator-(){

    x = -x;

    y = -y;

    z = -z;

}

int main(){

    space S;

    S.getdata(10,-20,30);

    cout << "S : ";

    S.display();

    -S;

    cout << "\n-S : ";

    S.display();

    return 0;

}

**Output :**

S : x = 10 Y = -20 Z =30

-S : x = -10 Y = 20 Z =-30

/\*

Vinay

Roll No.- 7118

Program-28 : Binary Operator Overloading

\*/

#include <iostream>

using namespace std;

class complex

{

    float real, imag;

public:

    complex()

    {

        real = imag = 0;

    }

    void read()

    {

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

        cin >> real >> imag;

    }

    complex operator+(complex a)

    {

        complex temp;

        temp.real = real + a.real;

        temp.imag = imag + a.imag;

        return temp;

    }

    void show()

    {

        if (imag > 0)

            cout << real << "+"

                 << "i" << imag;

    }

};

int main()

{

    complex c1, c2, c3;

    c1.read();

    c2.read();

    c3 = c1 + c2;

    cout << "\nC1 : ";c1.show();

    cout << "\nC2 : ";c2.show();

    cout << "\nC3 : ";c3.show();

    return 0;

}

**Output :**

Enter the numbers :

12

12

Enter the numbers :

10

10

C1 : 12+i12

C2 : 10+i10

C3 : 22+i22

/\*

Vinay

Roll No.- 7118

Program 29 : Conversion from user to basic

\*/

#include<iostream>

using namespace std;

class sample{

    int hrs, min;

    public:

    sample(int x,int y){

        hrs = x;

        min = y;

    }

    operator int(){

        return(hrs\*60+min);

    }

};

int main(){

    int h,m,duration;

    h=2,m=40;

    sample s(h,m);

    duration=s;

    cout << duration;

    }

**Output :**

160

/\*

Vinay

Roll No.- 7118

Program 30 : Conversion from basic to userdefined

\*/

#include<iostream>

using namespace std;

class sample{

    int hours,mins;

    public:

    sample(){

        hours = 0;

        mins = 0;

    }

    void display(){

        cout << hours << " "  << mins;

    }

    sample(int t){

        hours = t/60;

        mins = t%60;

    }

};

int main(){

    sample s;

    int x = 200;

    s = x;

    s.display();

    return 0;

}

**Output :**

3 20

/\*

Vinay

Roll No.- 7118

Program 31 : Conversion from user defined to userdefined

\*/

#include<iostream>

using namespace std;

class product

{

    public:

    int m, n;

    void setdata(int x, int y)

    {

        m=x;

        n=y;

    }

};

class item

{

    private:

    int a, b;

    public:

    void showdata()

    {

        cout<<a<<b;

    }

    item()

    {

        a=b=0;

    }

    item(product P)

    {

        a=P.m;

        b=P.n;

    }

};

int main()

{

    item i1;

    product p1;

    p1.setdata(3,5);

    i1=p1;

    i1.showdata();

    return 0;

}

**Output :**

35

/\*

Vinay

Roll No.- 7118

Program-32 : Single Inheritance

\*/

#include<iostream>

using namespace std;

class B{

    int a;

    public:

    int b;

    void set\_ab();

    int get\_a(void);

    void show\_a(void);

};

class D : public B{

    int c;

    public:

    void mul(void);

    void display(void);

};

void B :: set\_ab(void){

    a = 5;

    b = 10;

}

int B :: get\_a(){

   return a;

}

void B :: show\_a(){

    cout << "a = " << a << "\n";

}

void D :: mul(){

    c = b\*get\_a();

}

void D :: display(){

    cout << "a = " << get\_a() << "\n";

    cout << "b = " << b << "\n";

    cout << "c = " << c << "\n";

}

int main(){

    D d;

    d.set\_ab();

    d.mul();

    d.show\_a();

    d.display();

    d.b = 20;

    d.mul();

    d.display();

    return 0;

}

**Output :**

a = 5

a = 5

b = 10

c = 50

a = 5

b = 20

c = 100

/\*

Vinay

Roll No.- 7118

Program-33 : Multilevel inheritance

\*/

#include <iostream>

using namespace std;

class student{

    protected:

    int roll\_number;

    public:

    void get\_number(int);

    void put\_number(void);

};

void student :: get\_number(int a){

    roll\_number = a;

}

void student :: put\_number(){

    cout << "Roll Number: " << roll\_number << "\n";

}

class test : public student{

    protected:

    float sub1;

    float sub2;

    public:

    void get\_marks(float, float);

    void put\_marks(void);

};

void test :: get\_marks(float x,float y){

    sub1 = x;

    sub2 = y;

}

void test :: put\_marks(){

    cout << "Marks in Sub1 = " << sub1 << "\n";

    cout << "Marks in Sub2 = " << sub2 << "\n";

}

class result : public test{

    float total;

    public:

    void display(void);

};

void result :: display(void){

    total =  sub1 +sub2;

    put\_number();

    put\_marks();

    cout << "Total = " << total << endl;

}

int main(){

    result student1;

    student1.get\_number(111);

    student1.get\_marks(75.0,59.5);

    student1.display();

    return 0;

}

**Output :**

Roll Number: 111

Marks in Sub1 = 75

Marks in Sub2 = 59.5

Total = 134.5

/\*

Vinay

Roll No.- 7118

Program 34 : Multiple Inheritance

\*/

#include<iostream>

using namespace std;

class M{

    protected:

    int m;

    public:

    void get\_m(int);

};

class N{

    protected:

    int n;

    public:

    void get\_n(int);

};

class P : public M, public N{

    public:

    void display(void);

};

void M :: get\_m(int x){

    m = x;

}

void N :: get\_n(int y){

    n = y;

}

void P :: display(void){

    cout << "m = " << m << "\n";

    cout << "n = " << n << "\n";

    cout << "m\*n =" << m\*n << "\n";

}

int main(){

    P p;

    p.get\_m(10);

    p.get\_n(20);

    p.display();

    return 0;

}

**Output :**

m = 10

n = 20

m\*n =200

/\*

Vinay

Roll No.- 7118

Program-35 : Program to resolve inheritance ambiguity

\*/

#include<iostream>

using namespace std;

// Base class A

class A {

    public:

    void func() {

        cout << " I am in class A" << endl;

    }

};

// Base class B

class B {

    public:

    void func() {

        cout << " I am in class B" << endl;

    }

};

// Derived class C

class C: public A, public B {

};

int main() {

    C obj;

// Calling function func() in class A

    obj.A::func();

    // Calling function func() in class B

    obj.B::func();

    return 0;

}

**Output :**

I am in class A

I am in class B

/\*

Vinay

Roll No.- 7118

Program 36 : Hybrid Inheritance

\*/

#include<iostream>

using namespace std;

class student{

    protected:

    int roll\_number;

    public:

    void get\_number(int a){

        roll\_number = a;

    }

    void put\_number(){

        cout << "Roll no " << roll\_number << endl;

    }

};

class test : public student{

    protected:

    float part1, part2;

    public:

    void get\_marks(float x,float y){

        part1 = x;

        part2 = y;

    }

    void put\_marks(){

        cout << "Marks Obtained: " << endl;

        cout << "Part1 =  " << part1 << endl;

        cout << "Part2 = " << part2 << endl;

    }

};

class sports{

    protected:

    float score;

    public:

    void get\_score(float s){

        score = s;

    }

    void put\_score(){

        cout << "Sports wt : " << score << "\n\n";

    }

};

class result : public test, public sports{

    float total;

    public:

    void display();

};

void result :: display(){

    total = part1 + part2 + score;

    put\_number();

    put\_marks();

    put\_score();

    cout << "Total score : " << total << endl;

}

int main(){

    result student\_1;

    student\_1.get\_number(1234);

    student\_1.get\_marks(27.5,33.0);

    student\_1.get\_score(6.0);

    student\_1.display();

    return 0;

}

**Output :**

Roll no 1234

Marks Obtained:

Part1 = 27.5

Part2 = 33

Sports wt : 6

Total score : 66.5

/\*

Vinay

Roll No.- 7118

Program-37 : Virtual base class

\*/

#include<iostream>

using namespace std;

class student{

    protected:

    int roll\_number;

    public:

    void get\_number(int a){

        roll\_number = a;

    }

    void put\_number(){

        cout << "Roll no " << roll\_number << endl;

    }

};

class test : virtual public student{

    protected:

    float part1,part2;

    public:

    void get\_marks(float x,float y){

        part1 = x;

        part2 = y;

    }

    void put\_marks(){

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

        cout << "Part1 = " << part1 <<endl;

        cout << "Part2 = " << part2 <<endl;

    }

};

class sports : public virtual student{

    protected:

    float score;

    public:

    void get\_score(float s){

        score = s;

    }

    void put\_score(){

        cout << "Sports wt: " << score << "\n\n";

    }

};

class result : public test, public sports{

    float total;

    public:

    void display();

};

void result :: display(){

    total = part1 + part2 + score;

    put\_number();

    put\_marks();

    put\_score();

    cout << "Total score : " << total << endl;

}

int main(){

    result student\_1;

    student\_1.get\_number(678);

    student\_1.get\_marks(30.5,25.5);

    student\_1.get\_score(7.0);

    student\_1.display();

    return 0;

}

**Output :**

Roll no 678

Marks obtained :

Part1 = 30.5

Part2 = 25.5

Sports wt: 7

Total score : 63