**Practice-1**

**Aim: WAP to add two objects using binary plus (+) operator overloading.**

**Program:**

#include<iostream>

using namespace std;

class Addition

{

private:

int x ;

public:

void setdata()

{

cin>>x;

}

void getdata()

{

cout << x <<endl;

}

Addition operator+(Addition n)

{

Addition sum;

sum.x=this->x+n.x;

return sum;

}

};

int main()

{

Addition a1,a2,a3;

cout << "Enter first value:" << endl;

a1.setdata();

cout << "Enter second value:" << endl;

a2.setdata();

a3=a1+a2;

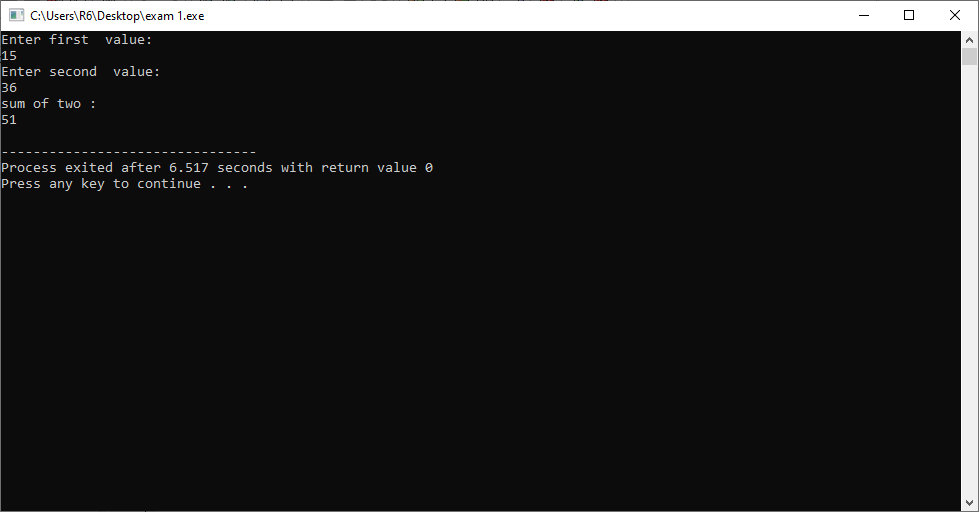
cout<< "sum of two :" << endl;

a3.getdata();

return 0;

}

**Output:**

****

**Practice-2**

**Aim:WAP to add two distances using binary plus (+) operator overloading.**

**Promgram:**

#include<iostream>

using namespace std;

class Distance

{

private:

int feet ;

int inch ;

public:

void setdata()

{

cout << "enter the feet :- "; cin>>feet;

cout << "enter the inch :- "; cin>>inch;

}

void getdata()

{

cout << "Feet:" << feet << "\t" << "Inches:" << inch << endl;

}

Distance operator+(Distance n)

{

Distance sum;

sum.inch=this->inch+n.inch;

sum.feet=this->feet+n.feet+(sum.inch/12);

sum.inch=sum.inch%12;

return sum;

}

};

int main()

{

Distance d1,d2,d3;

cout << "Enter first distance:" << endl;

d1.setdata();

cout << "Enter second distance:" << endl;

d2.setdata();

d3=d1+d2;

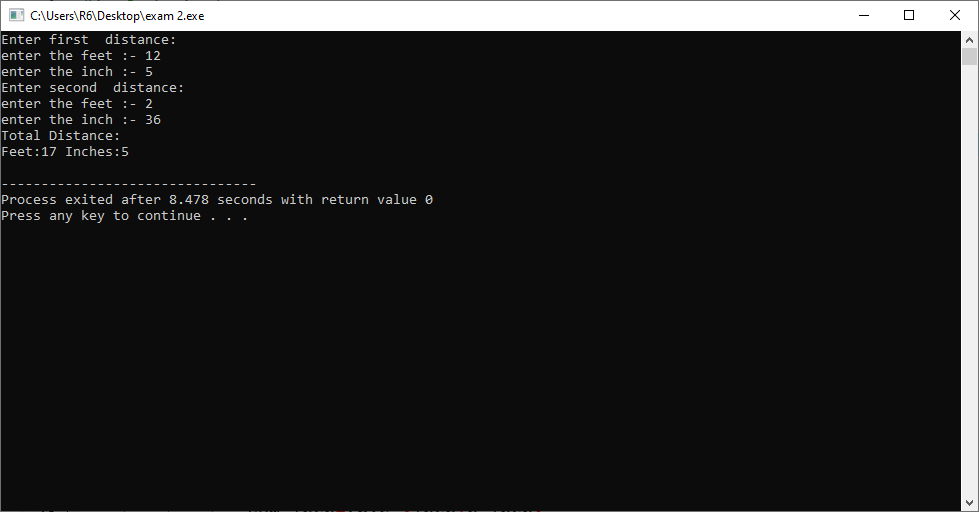
cout<< "Total Distance:" << endl;

d3.getdata();

return 0;

}

**Output:**

****