**Practical-1**

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

**Program:**

#include<iostream>

using namespace std;

class Object

{

private:

int num;

public:

void setData()

{

cout << "=> Enter number :- "; cin >> this->num;

}

void getData()

{

cout << endl << "\* Sum of both number :- " << this->num;

}

Object operator+(Object n)

{

Object s;

s.num = this->num + n.num;

return s;

}

};

int main()

{

Object o1,o2,o3;

o1.setData();

o2.setData();

o3 = o1 + o2;

o3.getData();

return 0;

}

**Output:**

****

**Practical-2**

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

**Program:**

#include<iostream>

using namespace std;

class Distance

{

private:

int feet,inch;

public:

void setData()

{

cout << "=> Enter Feet :- "; cin >> this->feet;

cout << "=> Enter Inch :- "; cin >> this->inch;

}

void getData()

{

cout << "=> Feet :- " << this->feet << endl;

cout << "=> Inch :- " << this->inch << endl;

}

Distance operator+(Distance n)

{

Distance temp;

temp.inch = (this->inch + n.inch)%12;

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

return temp;

}

};

int main()

{

Distance d1,d2,d3;

cout << "\* Enter distance value of a : " << endl;

d1.setData();

cout << "\* Enter distance value of b : " << endl;

d2.setData();

d3 = d1 + d2;

cout << endl << endl;

d3.getData();

return 0;

}

**Output:**

****