UNIVERSITY OF ENGINEERING AND TECHNOLOGY

(NAROWAL CAMPUS)

Object-Oriented Programming Lab Manual

Created by: Muhammad Abdullah

Registration Number: 2022-CS-525

Topics: Operator Overloading, Data Conversion,

& Returning Object from Operator Overloading

Lab Manual

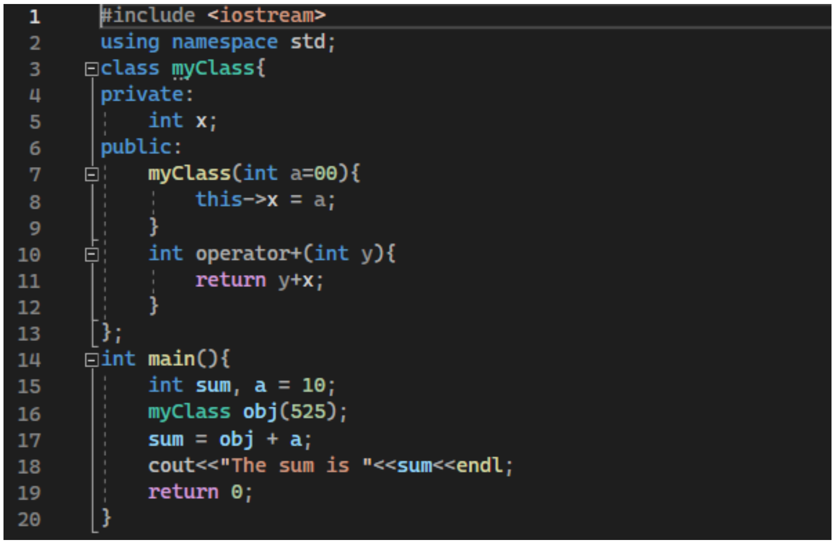
(Object-Oriented Programming Lab)

- ## Task 1:

Write an operator overloading program for the following cases:

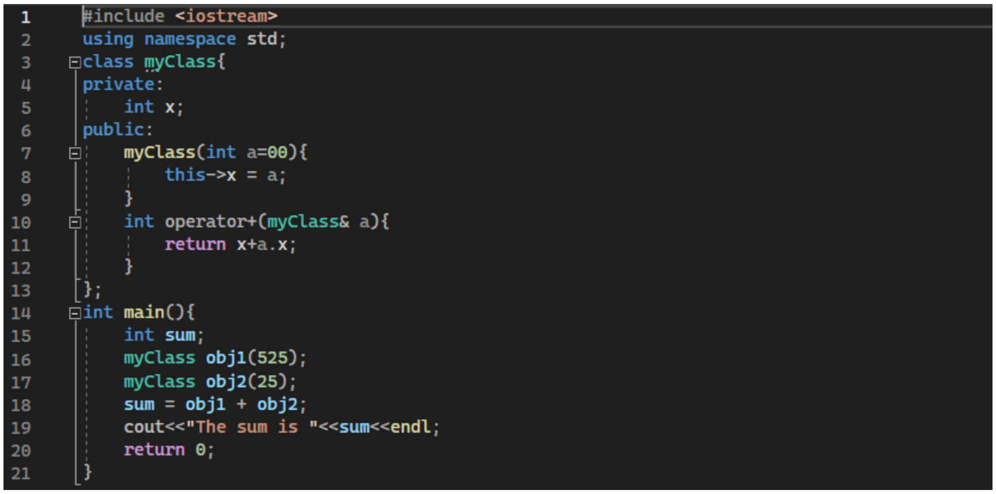
1. User-defined datatype sum primitive datatype assign to a primitive datatype.
2. User-defined datatype sum user-defined datatype assign to a primitive datatype.
3. User-defined datatype sum primitive datatype assign to a user-defined datatype.
4. User-defined datatype sum user-defined datatype assign to a user-defined datatype.

Program 1:



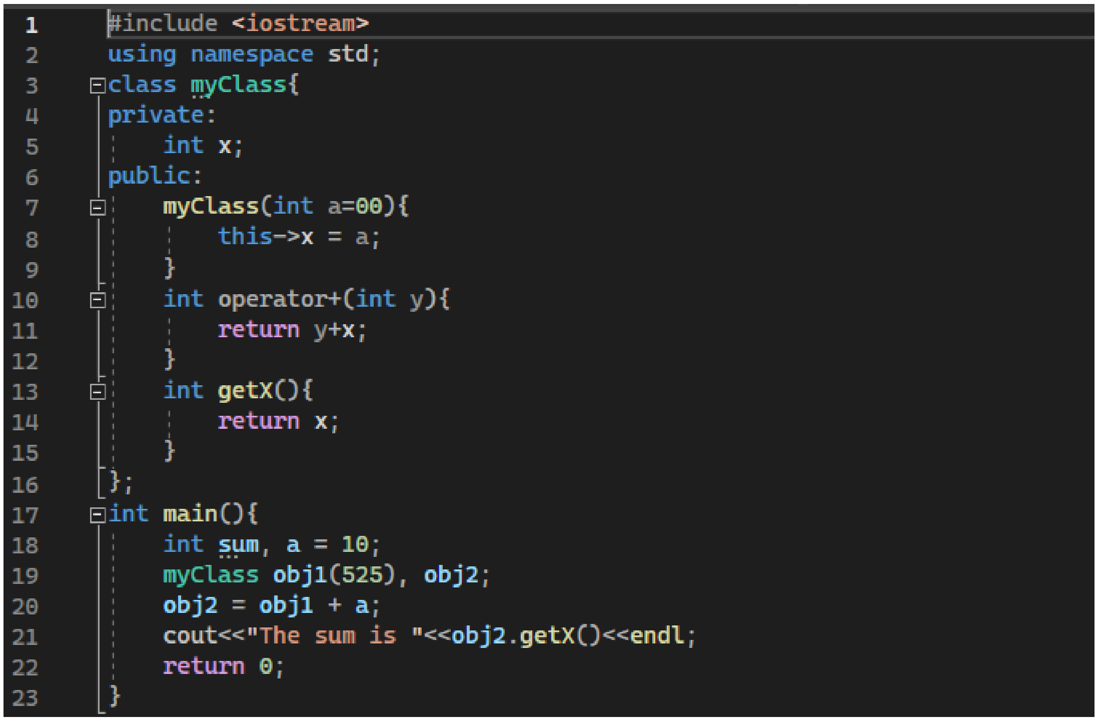
```cpp
#include <iostream>
using namespace std;
class myClass{
private:
    int x;
public:
    myClass(int a=00){
        this->x = a;
    }
    int operator+(int y){
        return y+x;
    }
};
int main(){
    int sum, a = 10;
    myClass obj(525);
    sum = obj + a;
    cout<<"The sum is "<<sum<<endl;
    return 0;
}
```

Program 2:



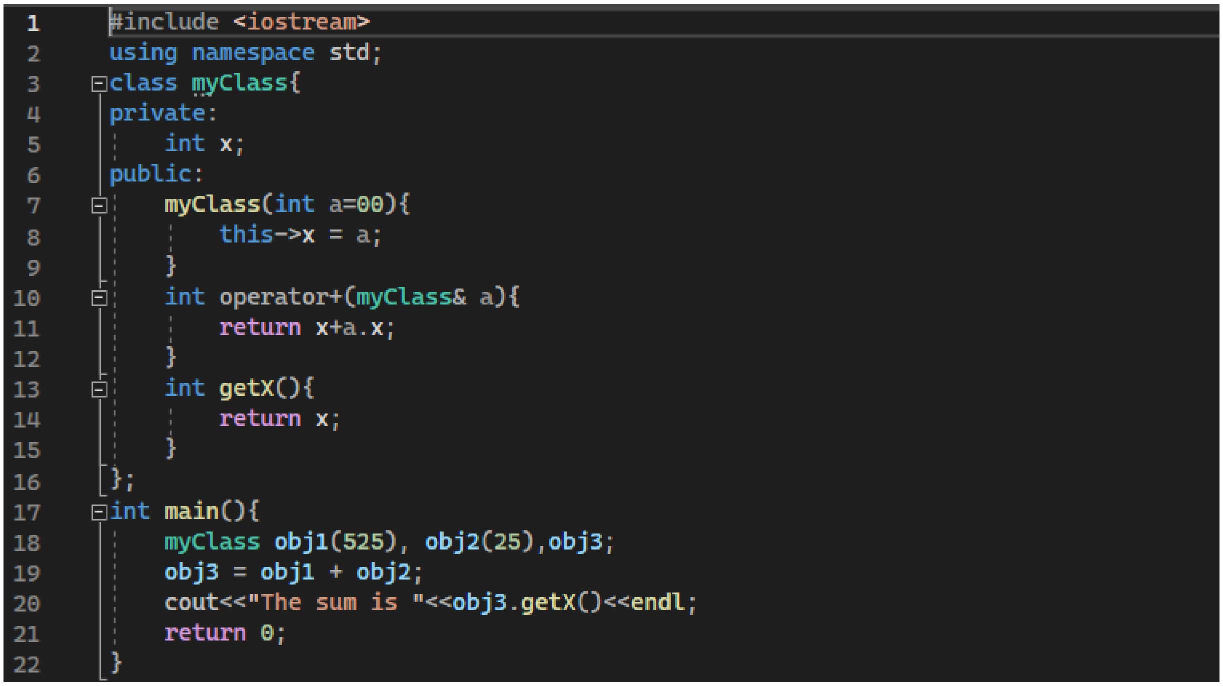
```cpp
#include <iostream>
using namespace std;
class myClass{
private:
    int x;
public:
    myClass(int a=00){
        this->x = a;
    }
    int operator+(myClass& a){
        return x+a.x;
    }
};
int main(){
    int sum;
    myClass obj1(525);
    myClass obj2(25);
    sum = obj1 + obj2;
    cout<<"The sum is "<<sum<<endl;
    return 0;
}
```

Program 3:



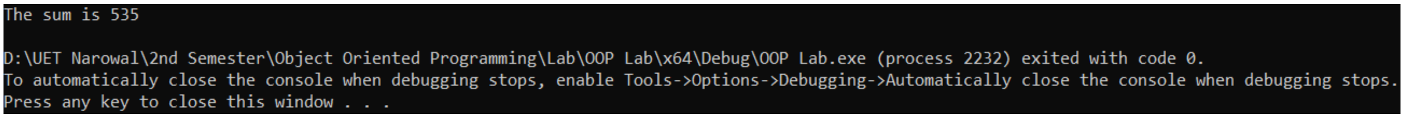
```cpp
#include <iostream>
using namespace std;
class myClass{
private:
    int x;
public:
    myClass(int a=00){
        this->x = a;
    }
    int operator+(int y){
        return y+x;
    }
    int getX(){
        return x;
    }
};
int main(){
    int sum, a = 10;
    myClass obj1(525), obj2;
    obj2 = obj1 + a;
    cout<<"The sum is "<<obj2.getX()<<endl;
    return 0;
}
```

Program 4:



```cpp
#include <iostream>
using namespace std;
class myClass{
private:
    int x;
public:
    myClass(int a=00){
        this->x = a;
    }
    int operator+(myClass& a){
        return x+a.x;
    }
    int getX(){
        return x;
    }
};
int main(){
    myClass obj1(525), obj2(25),obj3;
    obj3 = obj1 + obj2;
    cout<<"The sum is "<<obj3.getX()<<endl;
    return 0;
}
```

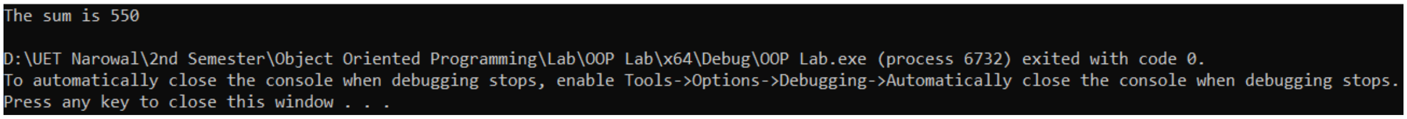
Output 1:



```
The sum is 535

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 2232) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

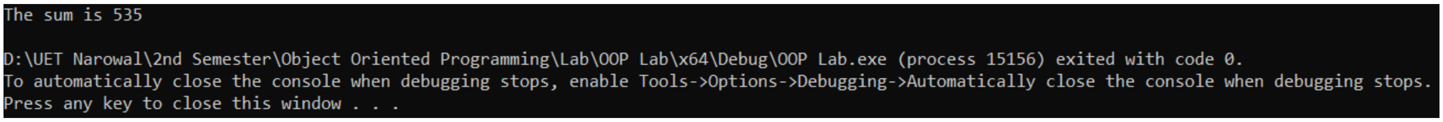
Output 2:



```
The sum is 550

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 6732) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

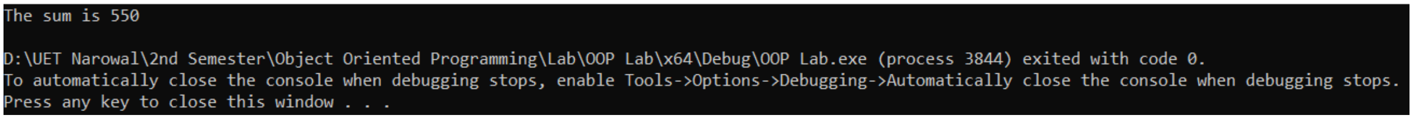
Output 3:



```
The sum is 535

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 15156) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Output 4:



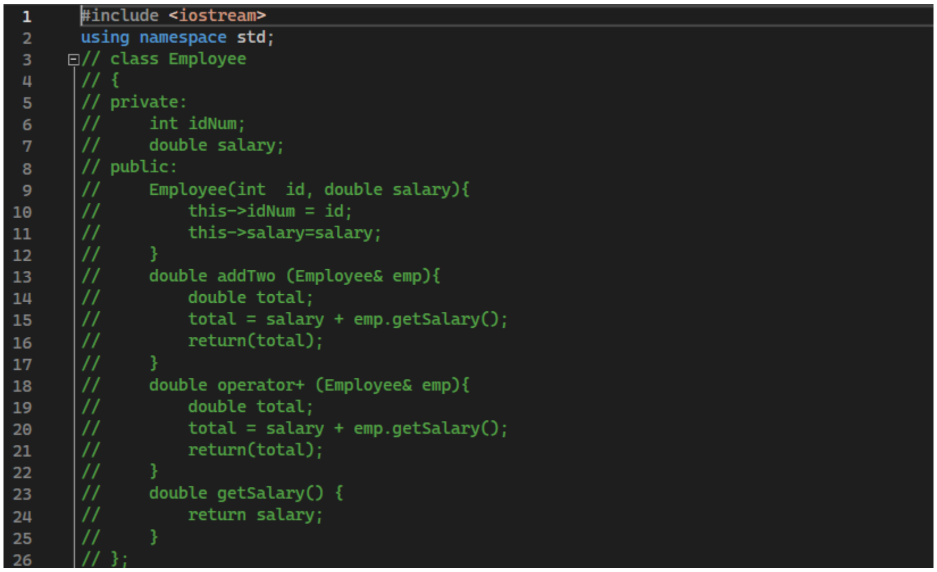
```
The sum is 550

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 3844) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

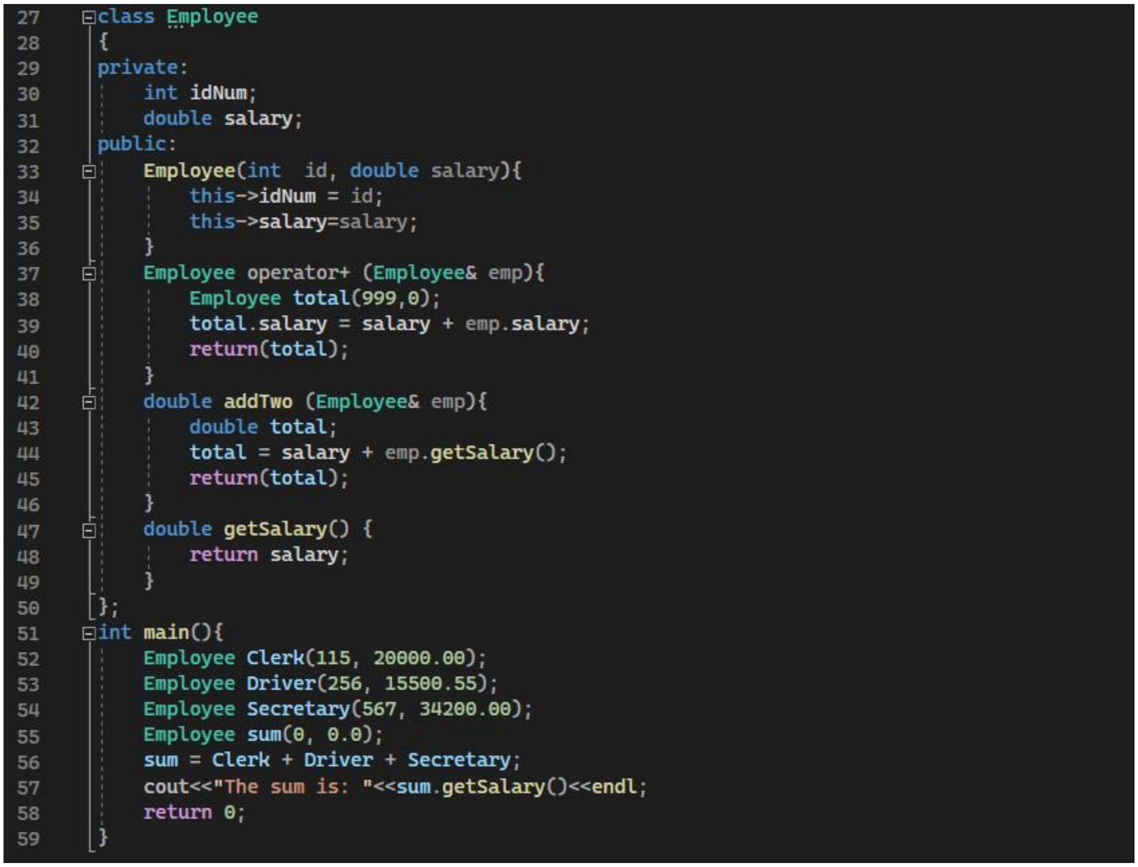
- ## Task 2,3,4:

Write a program of operator overloading that returns double and object. Also, demonstrate the sum of three objects and assign it to another object in a C++ program.

Program:

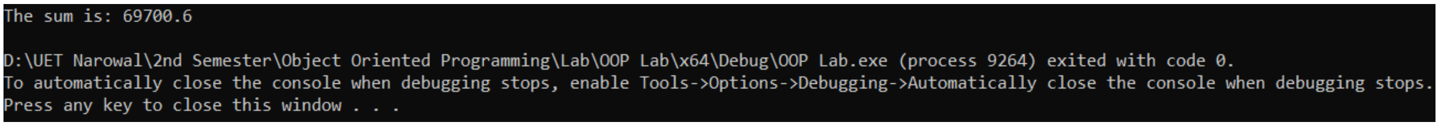


```cpp
#include <iostream>
using namespace std;
// class Employee
// {
// private:
//     int idNum;
//     double salary;
// public:
//     Employee(int  id, double salary){
//         this->idNum = id;
//         this->salary=salary;
//     }
//     double addTwo (Employee& emp){
//         double total;
//         total = salary + emp.getSalary();
//         return(total);
//     }
//     double operator+ (Employee& emp){
//         double total;
//         total = salary + emp.getSalary();
//         return(total);
//     }
//     double getSalary() {
//         return salary;
//     }
// };
```



```cpp
class Employee
{
private:
    int idNum;
    double salary;
public:
    Employee(int  id, double salary){
        this->idNum = id;
        this->salary=salary;
    }
    Employee operator+ (Employee& emp){
        Employee total(999,0);
        total.salary = salary + emp.salary;
        return(total);
    }
    double addTwo (Employee& emp){
        double total;
        total = salary + emp.getSalary();
        return(total);
    }
    double getSalary() {
        return salary;
    }
};
int main(){
    Employee Clerk(115, 20000.00);
    Employee Driver(256, 15500.55);
    Employee Secretary(567, 34200.00);
    Employee sum(0, 0.0);
    sum = Clerk + Driver + Secretary;
    cout<<"The sum is: "<<sum.getSalary()<<endl;
    return 0;
}
```

Output:



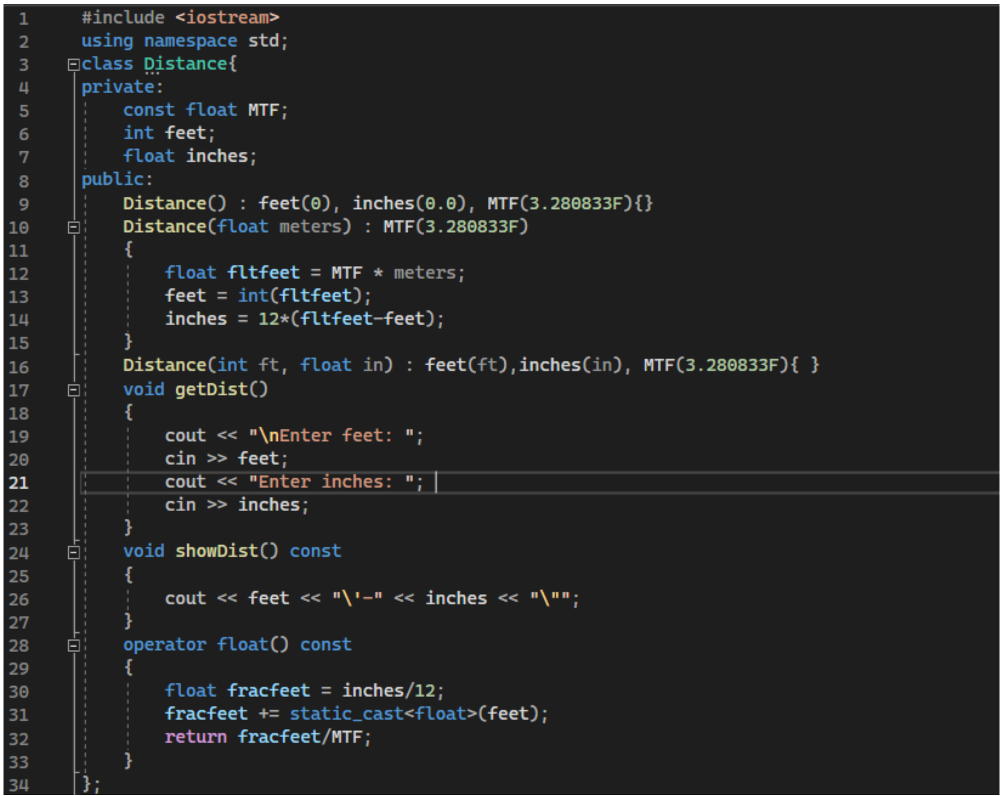
```
The sum is: 69700.6

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 9264) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

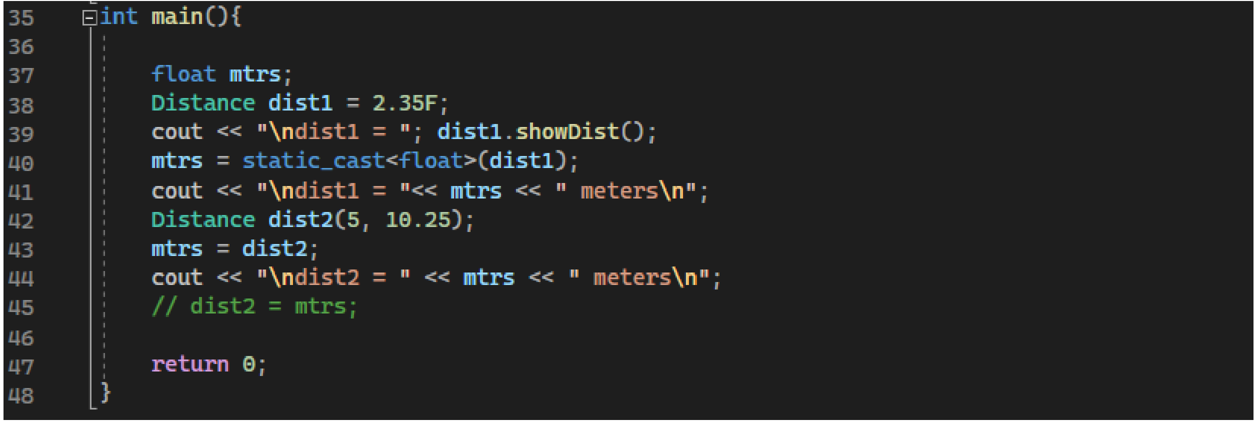
- ## Task 5:

Demonstrate the conversion of an object into a primitive datatype in a C++ program:

Program:



```cpp
#include <iostream>
using namespace std;
class Distance{
private:
    const float MTF;
    int feet;
    float inches;
public:
    Distance() : feet(0), inches(0.0), MTF(3.280833F){}
    Distance(float meters) : MTF(3.280833F)
    {
        float fltfeet = MTF * meters;
        feet = int(fltfeet);
        inches = 12*(fltfeet-feet);
    }
    Distance(int ft, float in) : feet(ft),inches(in), MTF(3.280833F){ }
    void getDist()
    {
        cout << "\nEnter feet: ";
        cin >> feet;
        cout << "Enter inches: ";
        cin >> inches;
    }
    void showDist() const
    {
        cout << feet << "\'-" << inches << "\"";
    }
    operator float() const
    {
        float fracfeet = inches/12;
        fracfeet += static_cast<float>(feet);
        return fracfeet/MTF;
    }
};
```

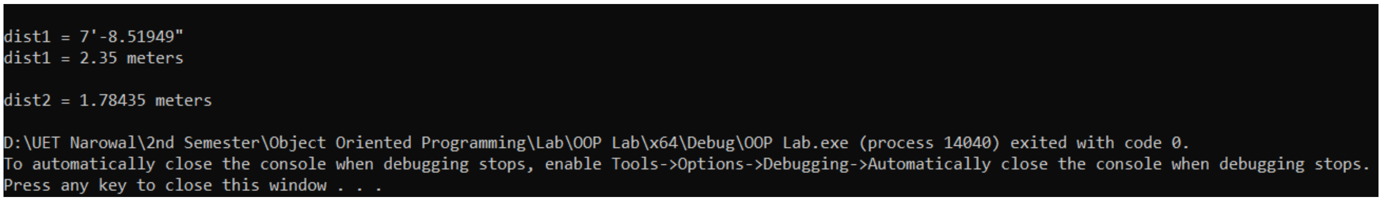


```cpp
int main(){

    float mtrs;
    Distance dist1 = 2.35F;
    cout << "\ndist1 = "; dist1.showDist();
    mtrs = static_cast<float>(dist1);
    cout << "\ndist1 ="<< mtrs << " meters\n";
    Distance dist2(5, 10.25);
    mtrs = dist2;
    cout << "\ndist2 = " << mtrs << " meters\n";
    // dist2 = mtrs;

    return 0;
}
```

Output:



```
dist1 = 7'-8.51949"
dist1 = 2.35 meters

dist2 = 1.78435 meters

D:\UET Narowal\2nd Semester\Object Oriented Programming\Lab\OOP Lab\x64\Debug\OOP Lab.exe (process 14040) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```
