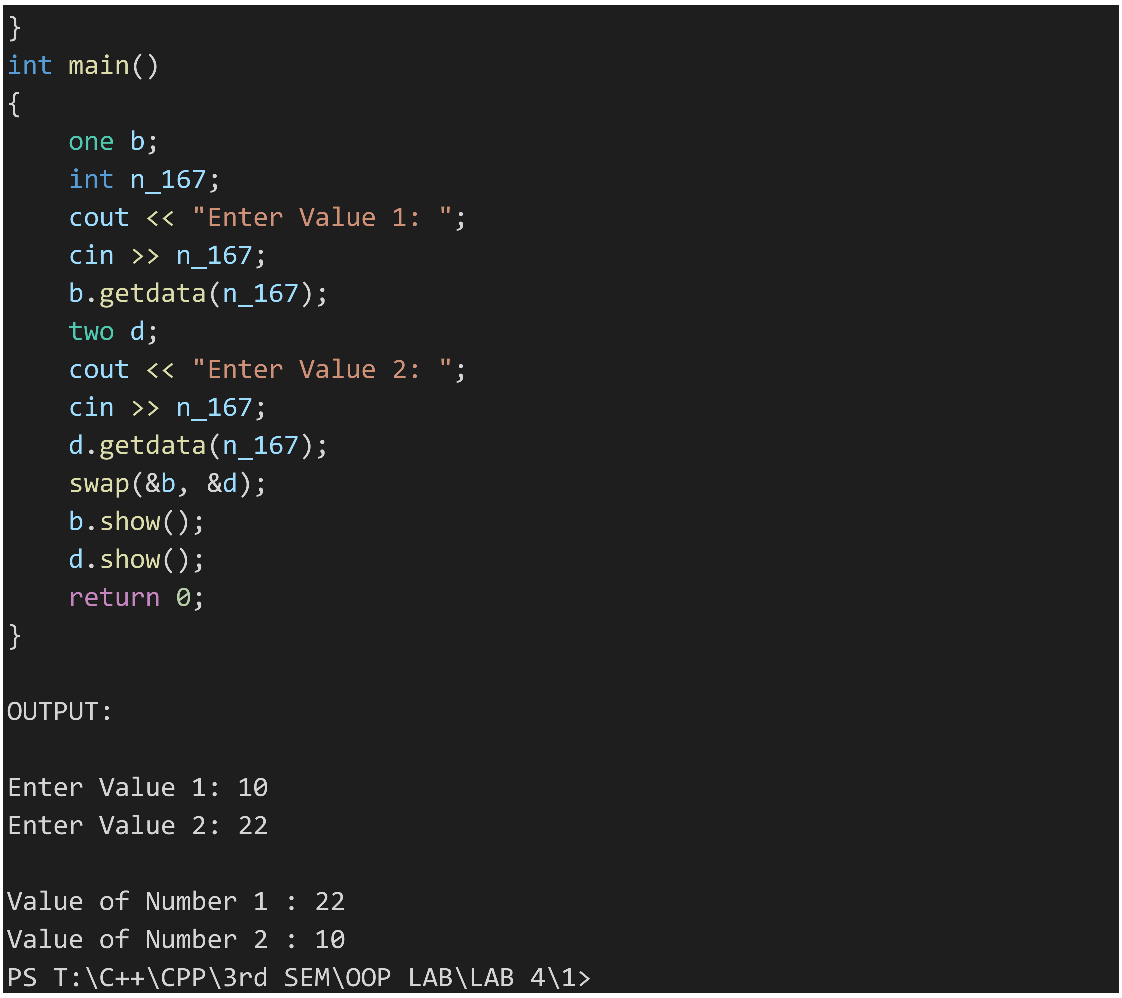
LAB 4

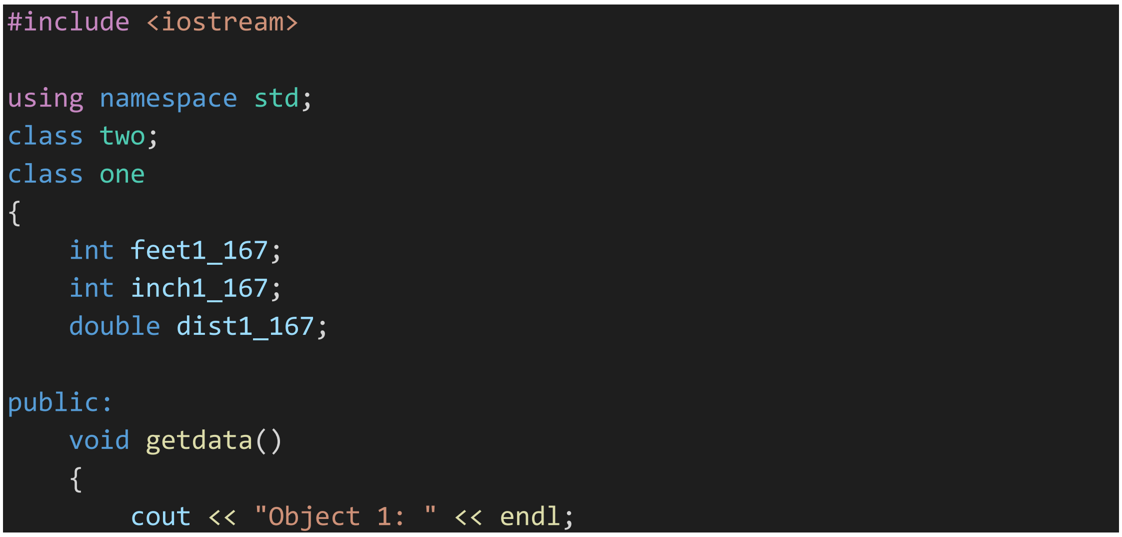
i. WAP to swap private data member of two different classes.
[The classes have no relation with each other].

```
#include <iostream>

using namespace std;

class two;
class one
{
    int num1_167;

public:
    void getdata(int a)
    {
        num1_167 = a;
    }
    void show()
    {
        cout << "\nValue of Number 1 : " << num1_167;
    }
    friend void swap(one *no1, two *no2);
};
class two
{
    int num2_167;

public:
    void getdata(int a)
    {
        num2_167 = a;
    }
    void show()
    {
        cout << "\nValue of Number 2 : " << num2_167;
    }
    friend void swap(one *no1, two *no2);
};
void swap(one *no1, two *no2)
{
    int no3;
    no3 = no1->num1_167;
    no1->num1_167 = no2->num2_167;
    no2->num2_167 = no3;
```

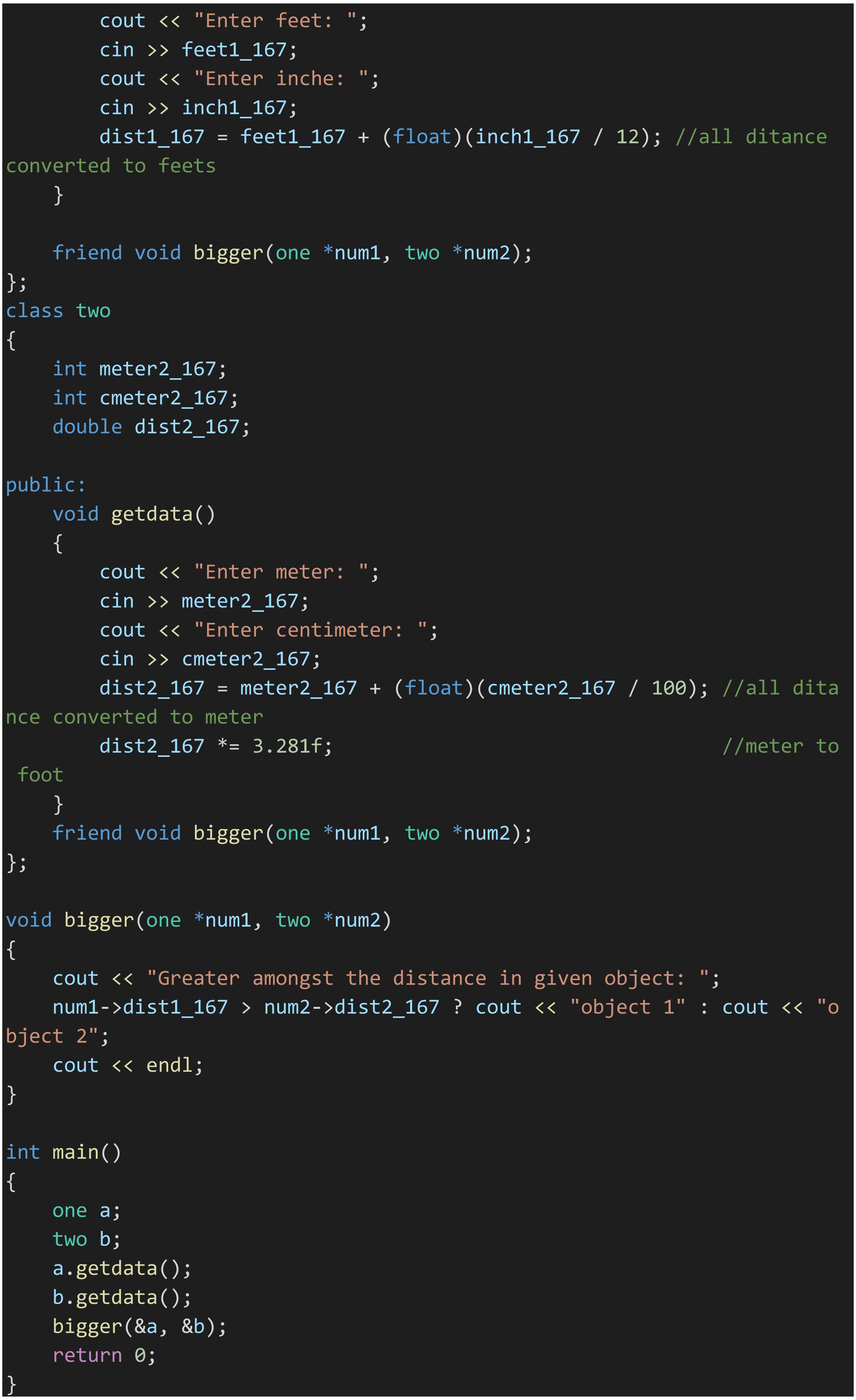


```
}
int main()
{
    one b;
    int n_167;
    cout << "Enter Value 1: ";
    cin >> n_167;
    b.getdata(n_167);
    two d;
    cout << "Enter Value 2: ";
    cin >> n_167;
    d.getdata(n_167);
    swap(&b, &d);
    b.show();
    d.show();
    return 0;
}

OUTPUT:

Enter Value 1: 10
Enter Value 2: 22

Value of Number 1 : 22
Value of Number 2 : 10
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\1>
```

ii. Create two classes which stores distance in feet, inches and meter, centimeter format respectively. Write a function which compares distance in object of these classes and displays the larger one.



```
#include <iostream>

using namespace std;
class two;
class one
{
    int feet1_167;
    int inch1_167;
    double dist1_167;

public:
    void getdata()
    {
        cout << "Object 1: " << endl;
```

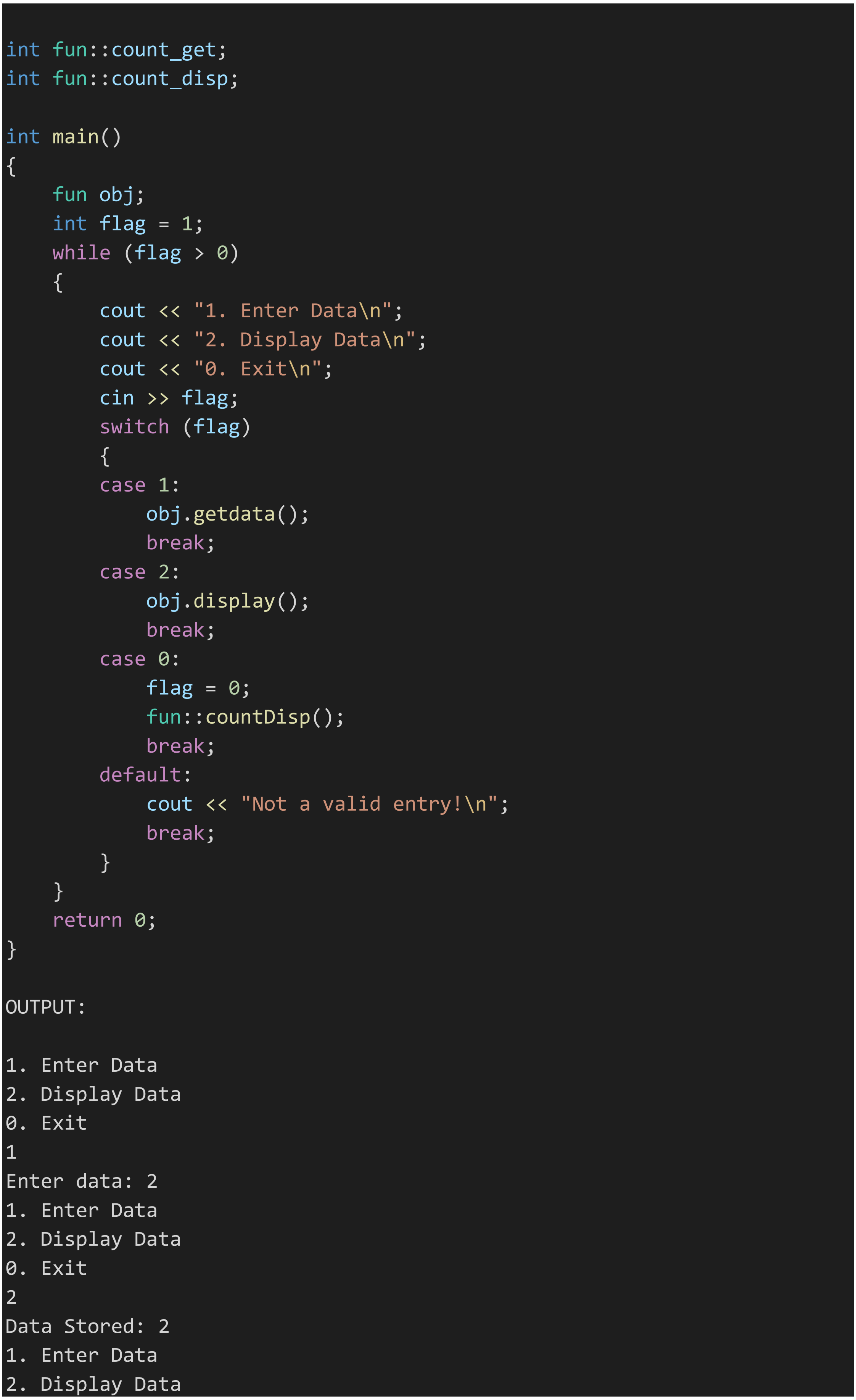


```
        cout << "Enter feet: ";
        cin >> feet1_167;
        cout << "Enter inche: ";
        cin >> inch1_167;
        dist1_167 = feet1_167 + (float)(inch1_167 / 12); //all ditance
converted to feets
    }

    friend void bigger(one *num1, two *num2);
};
class two
{
    int meter2_167;
    int cmeter2_167;
    double dist2_167;

public:
    void getdata()
    {
        cout << "Enter meter: ";
        cin >> meter2_167;
        cout << "Enter centimeter: ";
        cin >> cmeter2_167;
        dist2_167 = meter2_167 + (float)(cmeter2_167 / 100); //all dita
nce converted to meter
        dist2_167 *= 3.281f;                                  //meter to
 foot
    }
    friend void bigger(one *num1, two *num2);
};

void bigger(one *num1, two *num2)
{
    cout << "Greater amongst the distance in given object: ";
    num1->dist1_167 > num2->dist2_167 ? cout << "object 1" : cout << "o
bject 2";
    cout << endl;
}

int main()
{
    one a;
    two b;
    a.getdata();
    b.getdata();
    bigger(&a, &b);
    return 0;
}
```

```
OUTPUT:

Object 1:
Enter feet: 3
Enter inche: 13
Object 2:
Enter meter: 1
Enter centimeter: 150
Greater amongst the distance in given object: object 2
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\2>
```

iii. Create a class with an integer data member. Include functions for input and output in class. Count the number of times each function is called and display it.

```cpp
#include <iostream>
using namespace std;

class fun
{
    int n_167;
    static int count_get;
    static int count_disp;

public:
    void getdata()
    {
        ++count_get;
        cout << "Enter data: ";
        cin >> n_167;
    }
    void display()
    {
        ++count_disp;
        cout << "Data Stored: " << n_167 << endl;
    }
    static void countDisp()
    {
        cout << "Get Called: " << count_get << " times.\n";
        cout << "Display Called: " << count_disp << " times.\n";
    }
};
```



```
int fun::count_get;
int fun::count_disp;

int main()
{
    fun obj;
    int flag = 1;
    while (flag > 0)
    {
        cout << "1. Enter Data\n";
        cout << "2. Display Data\n";
        cout << "0. Exit\n";
        cin >> flag;
        switch (flag)
        {
        case 1:
            obj.getdata();
            break;
        case 2:
            obj.display();
            break;
        case 0:
            flag = 0;
            fun::countDisp();
            break;
        default:
            cout << "Not a valid entry!\n";
            break;
        }
    }
    return 0;
}
```

OUTPUT:

```
1. Enter Data
2. Display Data
0. Exit
1
Enter data: 2
1. Enter Data
2. Display Data
0. Exit
2
Data Stored: 2
1. Enter Data
2. Display Data
```

```
0. Exit
1
Enter data: 20
1. Enter Data
2. Display Data
0. Exit
1
Enter data: 1
1. Enter Data
2. Display Data
0. Exit
0
Get Called: 3 times.
Display Called: 1 times.
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\3>
```

iv. Create a class which stores name, roll number and total marks for a student. Input data for n students. Find the average marks scored by n students, store it as a data member of the class and display it using a function which may be called without object.

```cpp
#include <iostream>
#include <string>
using namespace std;

class student
{
    char name_167[20];
    int roll_167;
    int marks_167;
    static int avg;

public:
    void input_data()
    {
        cin.ignore();
        cout << "Enter name: ";
        cin.getline(name_167, 20);
        cout << "Enter Roll: ";
        cin >> roll_167;
        cout << "Enter Marks: ";
        cin >> marks_167;
    }
    friend void calc(student ar[], int n);
};
```

```
int student::avg;

void calc(student ar[], int n)
{
    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        sum += ar[i].marks_167;
    }
    student::avg = sum / n;
    cout << "Average Marks: " << student::avg;
}

int main()
{
    int n = 0;
    cout << "Enter No. of Students: ";
    cin >> n;
    student *obj = (student *)malloc(n * sizeof(student)); //DMA for array of class object
    if (!obj)
    {
        cout << "Insufficient Memory!";
    }
    else
    {
        for (int i = 0; i < n; i++)
        {
            obj[i].input_data();
        }
        calc(obj, n);
    }
    return 0;
}
```

```
OUTPUT:

1. Enter Student Data
0. Exit
1
Enter name: ANkit
Enter Roll: 2006167
Enter Marks: 200
1. Enter Student Data
0. Exit
1
Enter name: Arundhati
Enter Roll: 27112002
Enter Marks: 400
1. Enter Student Data
0. Exit
0
Average Marks: 300
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\4>
```

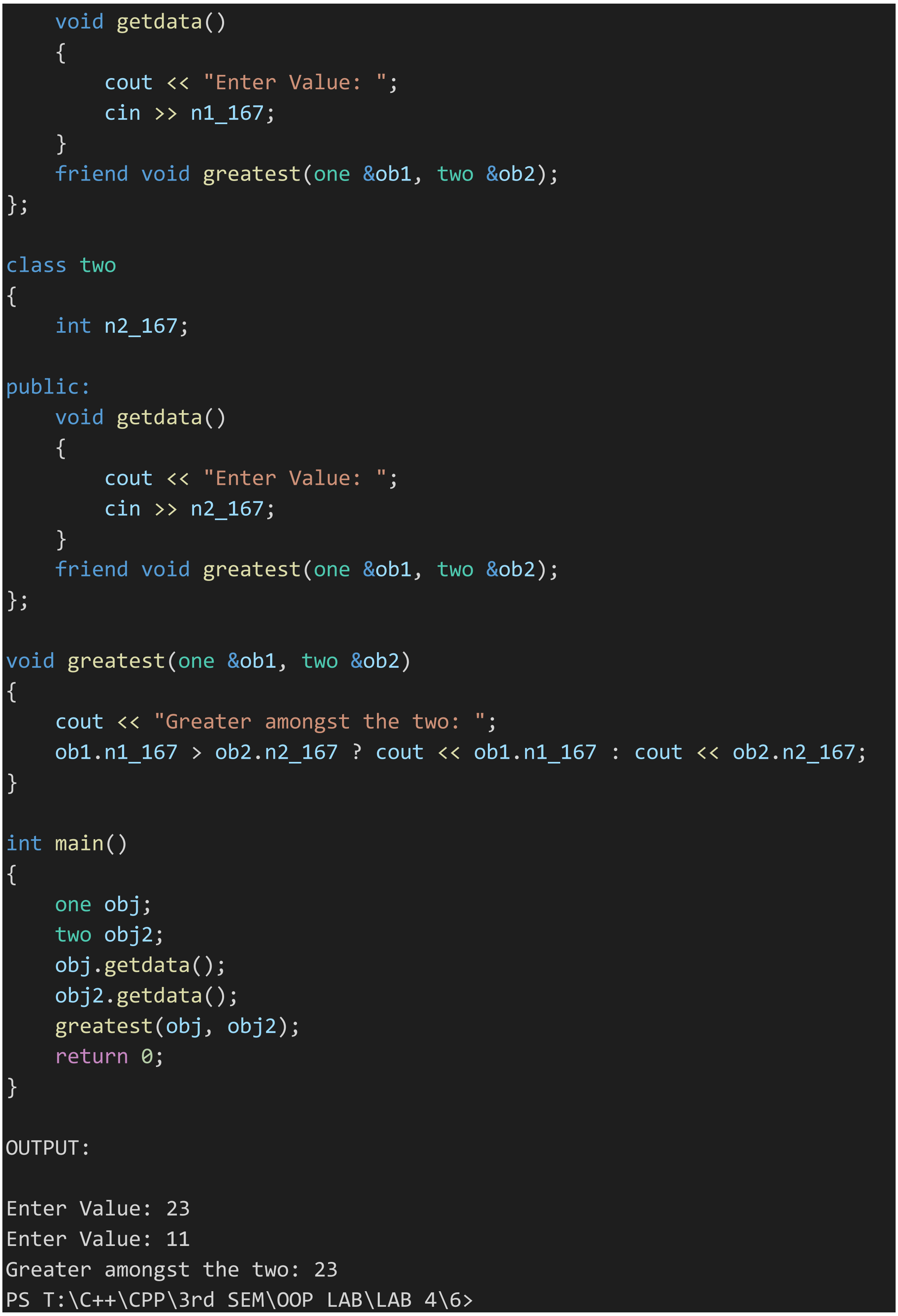
v. Create a class which stores name, author and price of a book. Store information for n number of books. Display information of all the books in a given price range using friend function in a tabular format.

```cpp
#include <iostream>
#include <string>
using namespace std;

class book
{
    char name_167[20];
    char author_167[20];
    int price_167;

public:
    void input_data()
    {
        cin.ignore();
        cout << "Enter name: ";
        cin.getline(name_167, 20);
        cout << "Enter Author: ";
        cin.getline(author_167, 20);
        cout << "Enter Price: ";
        cin >> price_167;
    }
    friend void display(book *p, int min, int max);
};

void display(book *p, int min, int max)
{
    cout << "Book in the given range: \n";
    if (p->price_167 >= min && p->price_167 <= max)
    {
        cout << "\nName: " << p->name_167 << endl;
        cout << "Author: " << p->author_167 << endl;
        cout << "Price: " << p->price_167 << endl;
    }
}

int main()
```

```
{
    int n;
    cout << "Enter no. of book: ";
    cin >> n;
    book obj[n];
    for (int i = 0; i < n; i++)
    {
        obj[i].input_data();
    }
    int min, max;
    cout << "Enter Min & Max Range: ";
    cin >> min >> max;
    for (int i = 0; i < n; i++)
    {
        display(&obj[i], min, max);
    }
}

OUTPUT:

Enter no. of book: 2
Enter name: ABC
Enter Author: abc
Enter Price: 200
Enter name: XYZ
Enter Author: xyz
Enter Price: 700
Enter Min & Max Range: 100 500
Book in the given range:

Name: ABC
Author: abc
Price: 200
Book in the given range:
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\5>
```

vi. .Write a program to find out the greatest between two numbers defined in two different classes by using friend function.

```
#include <iostream>
using namespace std;

class two;
class one
{
    int n1_167;

public:
```

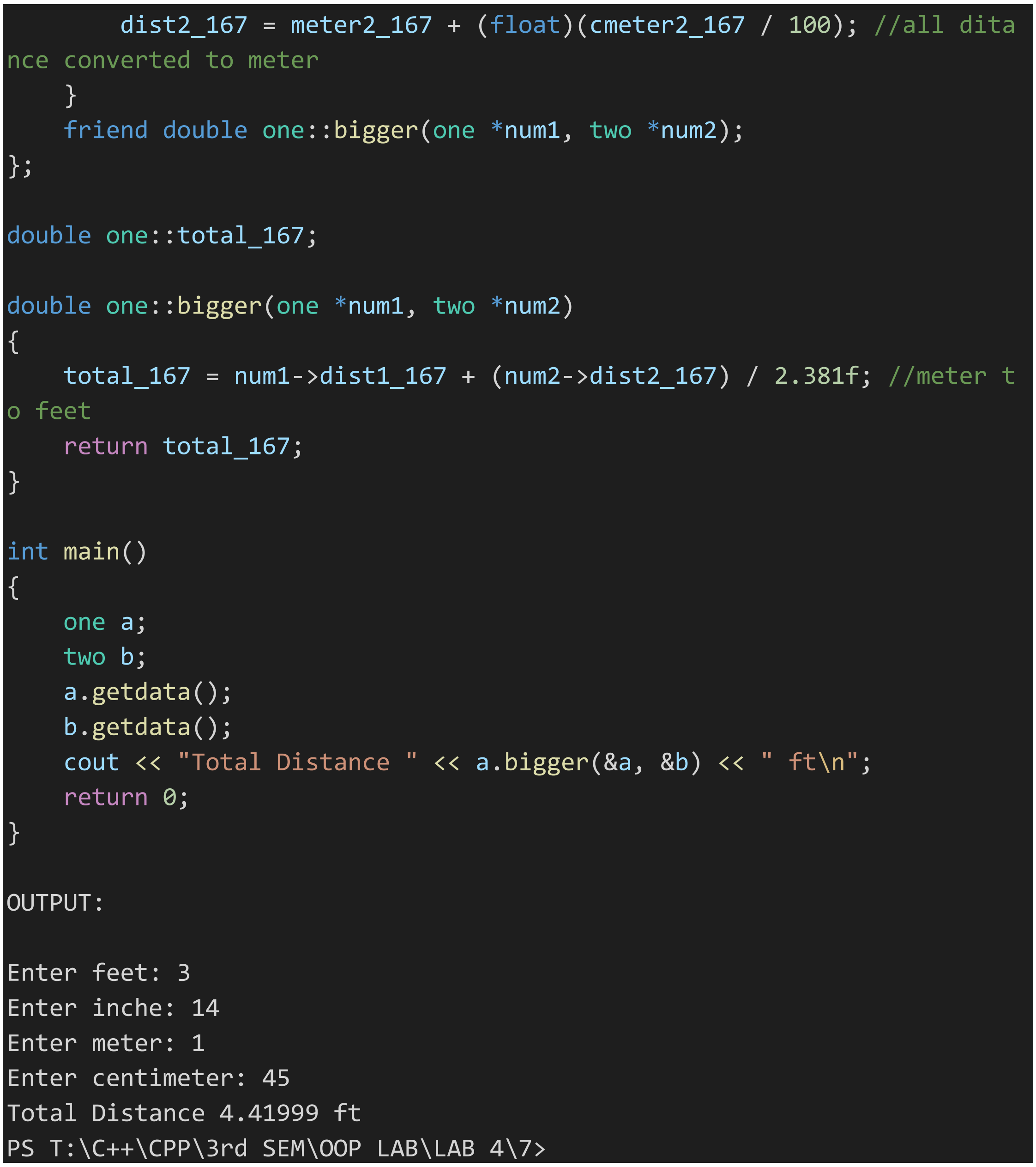


```
    void getdata()
    {
        cout << "Enter Value: ";
        cin >> n1_167;
    }
    friend void greatest(one &ob1, two &ob2);
};

class two
{
    int n2_167;

public:
    void getdata()
    {
        cout << "Enter Value: ";
        cin >> n2_167;
    }
    friend void greatest(one &ob1, two &ob2);
};

void greatest(one &ob1, two &ob2)
{
    cout << "Greater amongst the two: ";
    ob1.n1_167 > ob2.n2_167 ? cout << ob1.n1_167 : cout << ob2.n2_167;
}

int main()
{
    one obj;
    two obj2;
    obj.getdata();
    obj2.getdata();
    greatest(obj, obj2);
    return 0;
}

OUTPUT:

Enter Value: 23
Enter Value: 11
Greater amongst the two: 23
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\6>
```

vii. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of Dm with another object of DB. Use friend function to carry out the operation.

The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should in the format of feet and inches or meters and centimeters depending on the object on display.

```cpp
#include <iostream>

using namespace std;
class two;
class one
{
    int feet1_167;
    int inch1_167;
    double dist1_167;

    static double total_167;

public:
    void getdata()
    {
        cout << "Object 1: " << endl;
        cout << "Enter feet: ";
        cin >> feet1_167;
        cout << "Enter inche: ";
        cin >> inch1_167;
        dist1_167 = feet1_167 + (float)(inch1_167 / 12); //all ditance converted to feets
    }
    void display()
    {
        cout << "Total Distance: " << total_167 << endl;
    }
    double bigger(one *num1, two *num2);
};
class two
{
    int meter2_167;
    int cmeter2_167;
    int dist2_167;

public:
    void getdata()
    {
        cout << "Enter meter: ";
        cin >> meter2_167;
        cout << "Enter centimeter: ";
        cin >> cmeter2_167;
```



```
        dist2_167 = meter2_167 + (float)(cmeter2_167 / 100); //all dita
nce converted to meter
    }
    friend double one::bigger(one *num1, two *num2);
};

double one::total_167;

double one::bigger(one *num1, two *num2)
{
    total_167 = num1->dist1_167 + (num2->dist2_167) / 2.381f; //meter t
o feet
    return total_167;
}

int main()
{
    one a;
    two b;
    a.getdata();
    b.getdata();
    cout << "Total Distance " << a.bigger(&a, &b) << " ft\n";
    return 0;
}

OUTPUT:

Enter feet: 3
Enter inche: 14
Enter meter: 1
Enter centimeter: 45
Total Distance 4.41999 ft
PS T:\C++\CPP\3rd SEM\OOP LAB\LAB 4\7>
```