SESSIONAL TEST – 1

MCA
(Semester - II)

Name: Arzoo Khan

—

Roll No: 20MCA011

—

Paper Code: CSC26

—

Paper Title: Lab-III (OOP)

—

Date: 12-06-2021

—

Time: 2:30 PM – 3:30 PM

1. ***16.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 a friend function to carry out the addition 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 be in the format of feet and inches or metres and cenitmetres depending on the object on display..***

SOURCE CODE:

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

class MetreCentimetre;

class FeetInches
{
   int foot, inch;

public:
   FeetInches();
   FeetInches(int, int);
   void getdata();
   void show();
   friend void add(MetreCentimetre &, FeetInches &);
};

class MetreCentimetre
{
   int metre, centimetre;

public:
   MetreCentimetre();
   MetreCentimetre(int, int);
```

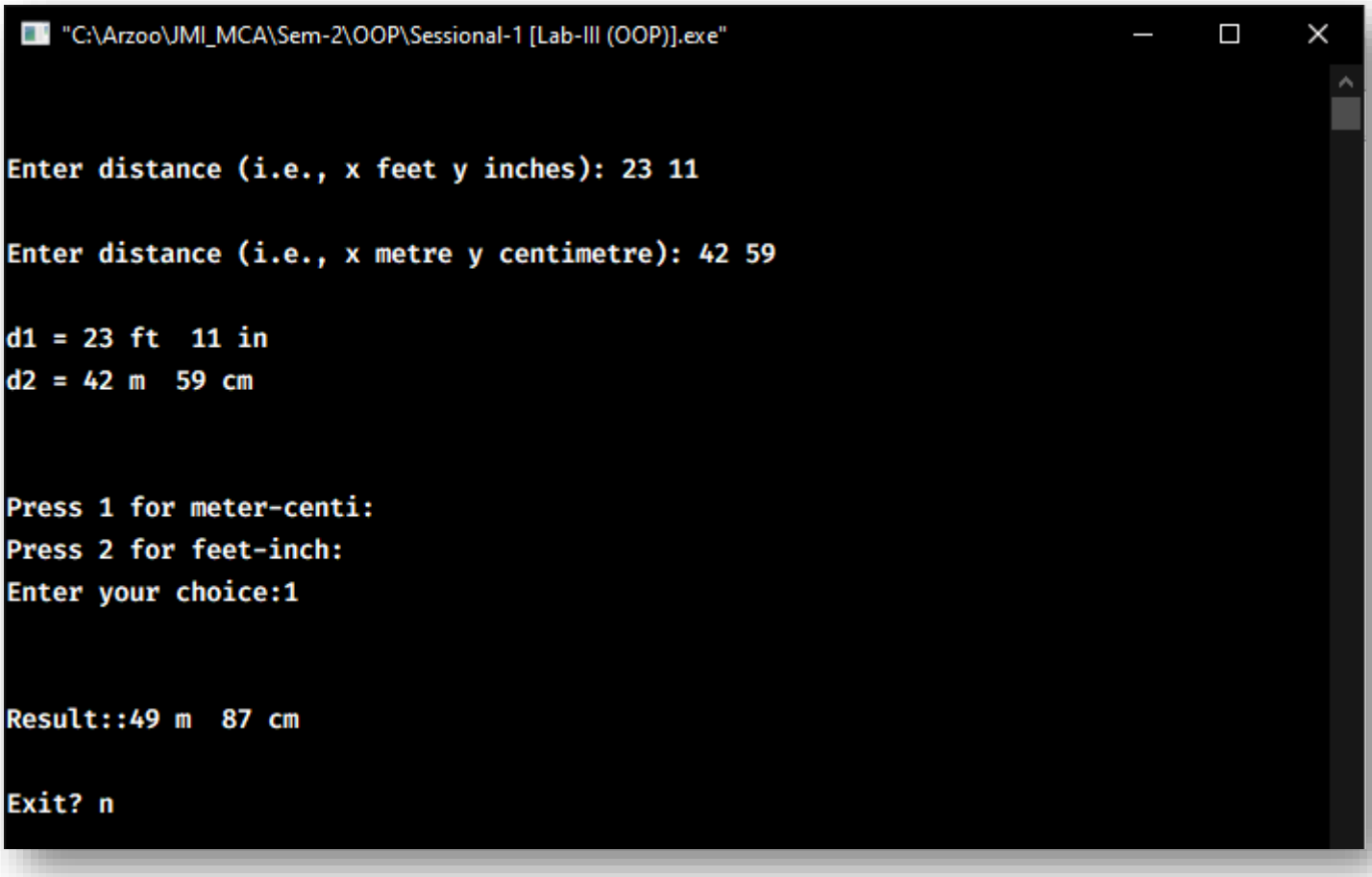
```
    void getdata();
    void show();
    friend void add(MetreCentimetre &, FeetInches &);
};

// Default Constructor
FeetInches ::FeetInches()
{
}

// Parameterized Constructor
FeetInches ::FeetInches(int ft, int in)
{
    foot = ft;
    inch = in;
}

// Member function to get data from the user
void FeetInches ::getdata()
{
    std ::cout << "\nEnter distance (i.e., x feet y inches): ";
    std ::cin >> foot >> inch;
    if (inch >= 12)
    {
        foot += inch / 12;
        inch %= 12;
    }
}

// Member function to show the content of the object of FeetInches Class
void FeetInches ::show()
{
    if (foot != 0)
        std ::cout << foot << " ft  ";
    if (inch != 0)
```

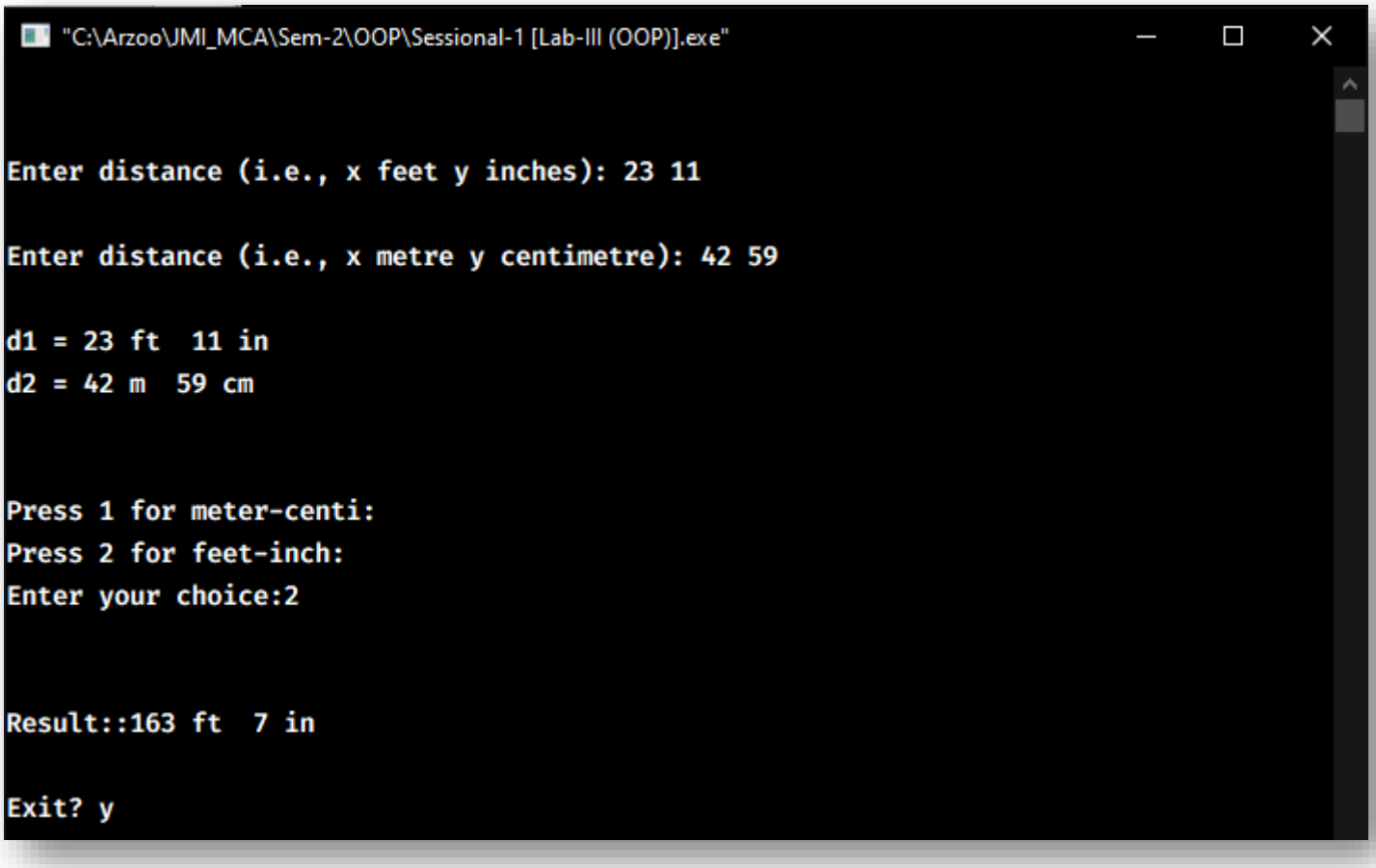
```
        std ::cout << inch << " in";
}
// Default Constructor
MetreCentimetre ::MetreCentimetre()
{
}

// Parameterized Constructor
MetreCentimetre ::MetreCentimetre(int m, int cm)
{
   metre = m;
   centimetre = cm;
}

// Member function to get data from the user
void MetreCentimetre ::getdata()
{
   std ::cout << "\nEnter distance (i.e., x metre y centimetre): ";
   std ::cin >> metre >> centimetre;
   if (centimetre >= 100)
   {
      metre += centimetre / 100;
      centimetre %= 100;
   }
}

// Member function to show the content of the object of FeetInches Class
void MetreCentimetre ::show()
{
   if (metre != 0)
      std ::cout << metre << " m  ";
   if (centimetre != 0)
      std ::cout << centimetre << " cm";
}
```

```
// Function to add two objects of class MetreCentimetre and FeetInches
void add(MetreCentimetre &d1, FeetInches &d2)
{
   int ch;
   cout << "\nPress 1 for meter-centi:";
   cout << "\nPress 2 for feet-inch:";
   cout << "\nEnter your choice:";
   cin >> ch;
   if (ch == 1)
   {
      MetreCentimetre d;
      int c = (d1.metre * 100 + d1.centimetre + d2.foot * 30.48 + d2.inch *
2.54);
      if (c >= 100)
      {
         d.metre = c / 100;
         d.centimetre = c % 100;
      }
      else
      {
         d.metre = 0;
         d.centimetre = c;
      }
      d.show();
   }
   else
   {
      FeetInches d;
      int i = (d1.metre * 39.37 + d1.centimetre * .3937008 + d2.foot * 12 +
d2.inch);
      if (i >= 12)
      {
         d.foot = i / 12;
         d.inch = i % 12;
      }
```

```
        else
        {
            d.foot = 0;
            d.inch = i;
        }
        d.show();
    }
}

//Driver Code
int main(void)
{
    FeetInches d1;
    MetreCentimetre d2;
    char ex;
    do
    {
        std ::cout << std::endl;
        d1.getdata();
        d2.getdata();

        std ::cout << "\nd1 = ";
        d1.show();
        std ::cout << "\n\nd2 = ";
        d2.show();

        add(d2, d1);

        std ::cout << "\n\nExit? ";
        std ::cin >> ex;
    } while (ex != 'y' && ex != 'Y');
    return 0;
}
```

OUTPUT:


"C:\Arzoo\JMI_MCA\Sem-2\OOP\Sessional-1 [Lab-III (OOP)].exe"
Enter distance (i.e., x feet y inches): 23 11
Enter distance (i.e., x metre y centimetre): 42 59
d1 = 23 ft 11 in
d2 = 42 m 59 cm
Press 1 for meter-centi:
Press 2 for feet-inch:
Enter your choice:1
Result::49 m 87 cm
Exit? n


"C:\Arzoo\JMI_MCA\Sem-2\OOP\Sessional-1 [Lab-III (OOP)].exe"
Enter distance (i.e., x feet y inches): 23 11
Enter distance (i.e., x metre y centimetre): 42 59
d1 = 23 ft 11 in
d2 = 42 m 59 cm
Press 1 for meter-centi:
Press 2 for feet-inch:
Enter your choice:2
Result::163 ft 7 in
Exit? y