

### **Assignment number - 3**

**Aim :** Create two classes DM and DB which stores values 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 addition operation

**Batch :** A-1

**Roll Number :** 04

---

### **Input Code :**

```
#include<iostream>
using namespace std;
class DB;
class DM {
    private:
        int meters;
        float centimeters;
    public:
        void setDistance(int m, float cm) {
            meters = m;
            centimeters = cm;
        }
        friend void addDistances(DM, DB);
};
class DB {
    private:
        int feet;
        float inches;
    public:
        void setDistance(int f, float in) {
            feet = f;
            inches = in;
        }
        friend void addDistances(DM, DB);
};
void addDistances(DM dmObj, DB dbObj) {
    float dmInCentimeters = dmObj.meters * 100 + dmObj.centimeters;
    float dbInInches = dbObj.feet * 12 + dbObj.inches;
    float totalDistance = dmInCentimeters + dbInInches;
    int totalMeters = static_cast<int>(totalDistance / 100);
    float remainingCentimeters = totalDistance - totalMeters * 100;
    cout<<"Total Distance : "<<totalMeters<<" meters and "<<remainingCentimeters<<"
centimeters"<<std::endl;
}
int main() {
    DM dmObj;
    DB dbObj;
    int meters;
    float centimeters;
    cout<<"Enter distance in meters and centimeters : \n";
    cin>>meters>>centimeters;
```

```
dmObj.setDistance(meters,centimeters);

int feet;
float inches;
cout<<"Enter distance in feet and inches :\n";
cin>>feet>>inches;
dbObj.setDistance(feet,inches);

addDistances(dmObj,dbObj);
return 0;
}
```

---

## **Output :**

```
Enter distance in meters and centimeters :
55
6
Enter distance in feet and inches :
23
7
Total Distance :57 meters and 89 centimeters
```