**/\* Program No. :**

**Aim : WAP to create two classes DM and DB which store the value of distances. DM stores distances in metres and centimetres and DB in feet and inches and add both the distances and shows the result in both formats using a friend function.**

**\*/**

#include<iostream.h>

#include<conio.h>

class db;

class dm

{

float metre,centimetre;

public:

dm(float);

dm(int);

dm(int,int);

void display() const;

friend void add(dm,db);

};

class db

{

float feet,inches;

public:

db(float);

db(int);

db(int,int);

void display() const;

friend void add(dm,db);

};

dm::dm(int a)

{

metre=a/100;

centimetre=a%100;

}

dm::dm(float a)

{

metre=(int)a;

centimetre=(a-metre)\*100;

}

dm::dm(int a,int b)

{

metre=a;

centimetre=b;

}

void dm::display() const

{

cout<<(int)metre<<" m and "<<(int)centimetre<<" cm";

}

db::db(int a)

{

feet=a/12;

inches=a%12;

}

db::db(float a)

{

feet=(int)a;

inches=(a-feet)\*12;

}

db::db(int a,int b)

{

feet=a;

inches=b;

}

void db::display() const

{

cout<<(int)feet<<" feet and "<<(int)inches<<" inches";

}

void add(dm ob1,db ob2)

{

cout<<"\n\n\t\tTotal distance : "

<<((ob1.metre\*100)+ob1.centimetre)+(((ob2.feet\*12)+ob2.inches)\*(30.48/12))

<<" cms";

cout<<"\n\n\t\tTotal distance : "

<<((ob1.metre+(ob1.centimetre/100))\*(3.2808))+(ob2.feet+(ob2.inches/12))

<<" feets";

}

void main()

{

clrscr();

dm m1(548);

db b1(74);

cout<<"\n\n\t\tDistance in DM : ";

m1.display();

cout<<"\n\n\t\tDistance in DB : ";

b1.display();

add(m1,b1);

getch();

}

**/\***

**Name : Rohit Aggarwal**

**Roll No. : 7CS-097**

**\*/**