1. Create a class Distance{feet, inches}: operations- default constructor, one-argument constructor (Distance(int), Distance(float)), two-argument constructor, copy constructor, input, display, add, convert to meter.(use operator overloading). Overload casting operators to convert class Distance to feet (float) or inches (integer) and to another class Meter {mt, cm}. Also implement >, >=, <, <=, <<, >> and != operator.

* #include<iostream>

using namespace std;

class Distance

{

private:

int feet,inches;

public:

void readDistance(void)

{

cout << "Enter feet: ";

cin >>feet;

cout << "Enter inches: ";

cin >>inches;

}

void dispDistance(void)

{

cout << "Feet:" << feet << "\t" << "Inches:" << inches << endl;

}

Distance operator>>( Distance &d)

{

Distance dis;

dis.inches>>d.feet;

return dis;

}

Distance operator<<( Distance &d)

{

Distance dos;

dos.inches<<d.feet;

return dos;

}

Distance operator+(Distance &dist1)

{

Distance tempD;

tempD.inches= inches + dist1.inches;

tempD.feet = feet + dist1.feet + (tempD.inches/12);

tempD.inches=tempD.inches%12;

return tempD;

}

void operator> (int)

{

if(feet>inches)

cout<<"feet is greater than inches";

}

void operator== (int)

{

if(inches==1/12);

cout<<"feet is equal to inches";

}

void operator!= (int)

{

if(inches!=1/12);

cout<<"feet is not equal to inch";

}

void operator< (int)

{

if(feet<inches)

cout<<"feet is lesser than inches";

}

void operator<= (int)

{

if(feet<=1/12)

cout<<"inch maximum value is 1";

}

void operator>= (int)

{

if(feet>=1/12)

cout<<"inch minimum value is 1";

}

};

class meter

{

private:

int mt,cm;

public:

void read(void)

{

cout << "Enter meter: ";

cin >>mt;

cout << "Enter centimeter: ";

cin >>cm;

}

void disp(void)

{

cout << "Meter:" << mt << "\t" << "Centimeter:" << cm << endl;

}

meter operator+(meter &dist1)

{

meter tempD;

tempD.cm= cm + dist1.cm;

tempD.mt = mt + dist1.mt + (tempD.cm/100);

tempD.cm=tempD.cm%100;

return tempD;

}

};

int main()

{

Distance D1,D2,D3;

cout << "Enter first distance:" << endl;

D1.readDistance();

cout << endl;

cout << "Enter second distance:" << endl;

D2.readDistance();

cout << endl;

D3=D1+D2;

cout << "Total Distance:" << endl;

D3.dispDistance();

meter a1,a2,a3;

cout << "Enter first distance:" << endl;

a1.read();

cout << endl;

cout << "Enter second distance:" << endl;

a2.read();

cout << endl;

a3=a1+a2;

cout << "Total Distance:" << endl;

a3.disp();

cout << endl

return 0;

}