```cpp
/*  Program: A5P4 - Templates
    Author: Tom Stutler
    Last Date Modified: 5/7/2015

    The intent of this program is grades.
*/

#include <iostream>

using namespace std;

const int MAX_SIZE = 5;

template <class T>
class list
{
private:
    T array[MAX_SIZE];

public:
    list () : array() {}
    list (const list& l) : array(l.array) {}

    void getList ()
    {
        for (int i=0; i<MAX_SIZE; i++) {
            cout << "\nEnter term " << i+1 << " for the list: ";
            cin >> array[i];
        }
    }
    void display ()
    {
        for (int i=0; i<MAX_SIZE; i++) {
            cout << "Term " << i+1 << " = " << array[i] << endl;
        }
    }
    void sortlist ()
    {
        T temp;

        for (int i=0; i<MAX_SIZE-1; i++) {
            for (int j=1; j<MAX_SIZE-i; j++) {
                if (array[j] < array[j-1]) {
                    temp = array[j];
                    array[j] = array[j-1];
                    array[j-1] = temp;
                }
            }
        }
    }
};

class Distance //English Distance class
{
private:
    int feet;
    float inches;

public:
    Distance() : feet(0), inches(0.0) //constructor (no args)
    { }
    Distance(int ft, float in) : feet(ft), inches(in)
    { } //constructor (two args)
    Distance( float fltfeet ) //constructor (one arg)
    { //convert float to Distance
        feet = int(fltfeet); //feet is integer part
        inches = 12*(fltfeet-feet); //inches is what's left
    }
    bool operator < (Distance) const; //compare distances
    friend istream& operator >> (istream& s, Distance& d);
    friend ostream& operator << (ostream& s, Distance& d);
};

template <class T>
void demo (T varFlag);

int main()
{
    int sel;
    bool end=false;
    int iflag=0;
    float fflag=0;
    char dflag=0;
    Distance Dflag;

    cout << "TEMPLATE DEMO PROGRAM\n";
    do{
        cout << "Enter list type (1=int 2=float 3=char 4=Distance 5=exit): ";
        cin >> sel;
        switch (sel)
        {
        case 1:
            demo(iflag);
            break;
        case 2:
            demo(fflag);
            break;
        case 3:
            demo(dflag);
            break;
        case 4:
            demo(Dflag);
            break;
        default:
            end=true;
            cout << "Bye...\n";
            break;
        }
    }while(!end);
    return 0;
}

template <class T>
void demo (T varFlag)
{
    list<T> list1, list2;

    cout << "\nDisplaying list1: \n";
    list1.display();
    list1.getList();
    list2 = list1;
    cout << "\nDisplaying list2: \n";
    list2.display();
    list1.sortlist();
    cout << "\nDisplaying list1 after sorting: \n";
    list1.display();
}

bool Distance::operator < (Distance d2) const //return the sum
{
    float bf1 = feet + inches/12;
    float bf2 = d2.feet + d2.inches/12;
    return (bf1 < bf2) ? true : false;
}
//--------------------------------------------------------------
istream& operator >> (istream& s, Distance& d) //get Distance
{ //from user
    cout << "\nEnter feet: "; s >> d.feet; //using
    cout << "Enter inches: "; s >> d.inches; //overloaded
    return s; //>> operator
}
//--------------------------------------------------------------
ostream& operator << (ostream& s, Distance& d) //display
{ //Distance
    s << d.feet << "\'-" << d.inches << '\"'; //using
    return s; //overloaded
} //<< operator
```