# Specialized Template Example we wrote in class

#include <iostream>  
#include <cmath>

using namespace std;

template <class T>  
class CompareIt  
{  
private:  
T a;  
T b;  
public:  
CompareIt(T x =0, T y = 0){a = x; b = y;}  
bool compared() {return (a == b ? true : false);}  
};

template <>  
class CompareIt<float>  
{  
private:  
float a;  
float b;  
float const SMALL = 0.0001;  
public:  
CompareIt(float x =0, float y = 0){a = x; b = y;}  
bool compared() {return (fabs(a - b)< SMALL ? true : false);}

};

int main()  
{

CompareIt<float> a(2.0,2.0);  
CompareIt<int> b(1,1);  
CompareIt<string> c("Hello", "Bye");

cout << a.compared() << endl;  
cout << b.compared() << endl;  
cout << c.compared() << endl;  
}