Week 14-15

Q5. Program with member function templates

#include <iostream>

using namespace std;

template <typename T>

class Box {

private:

T length;

T width;

T height;

public:

Box(T l, T w, T h) : length(l), width(w), height(h) {}

template <typename U>

Box(const Box<U>& other) {

length = static\_cast<T>(other.length);

width = static\_cast<T>(other.width);

height = static\_cast<T>(other.height);

}

template <typename U>

Box<T> add(const Box<U>& other) {

T l = length + static\_cast<T>(other.length);

T w = width + static\_cast<T>(other.width);

T h = height + static\_cast<T>(other.height);

return Box<T>(l, w, h);

}

void display() {

cout << "Length: " << length << ", Width: " << width << ", Height: " << height << endl;

}

};

int main() {

Box<int> intBox(10, 20, 30);

intBox.display();

Box<double> doubleBox(intBox);

doubleBox.display();

Box<int> resultBox = intBox.add(doubleBox);

resultBox.display();

return 0;

}