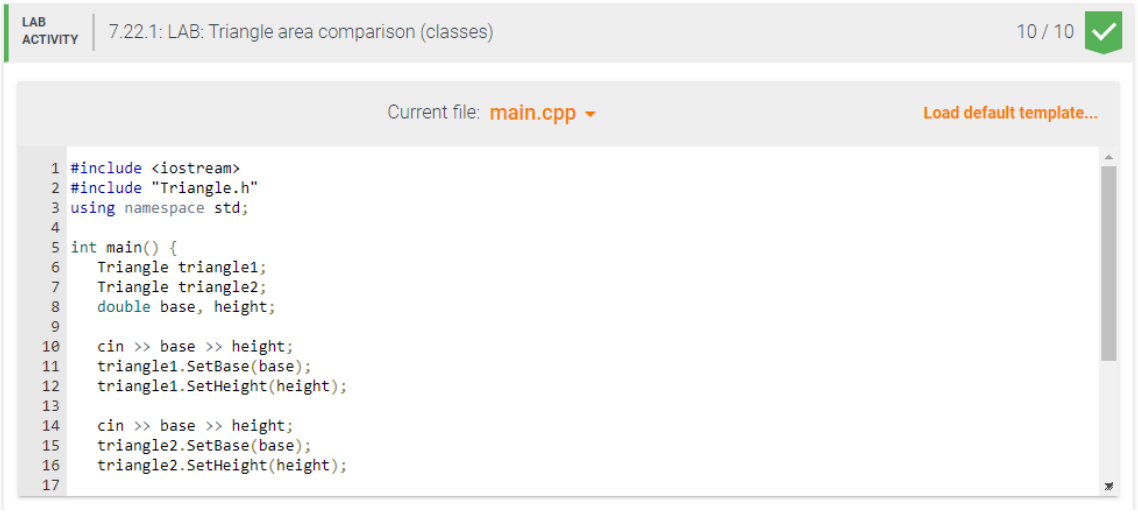
CSCI 140 PA 6 Submission

Due Date: 4/7/23

Name: Ali Mortada

Exercise 1 – **zyBook 7.22 LAB: Triangle area comparison** -- check if completely done in zyBook __X__ ; otherwise, discuss issues below
Include a screenshot of current status/score



Exercise 2 – **zyBook 7.23 LAB: Car value** -- check if completely done in zyBook __X__ ; otherwise, discuss issues below
Include a screenshot of current status/score

Exercise 3 – **zyBook 7.27 LAB*: Warm up: Online shopping cart (Part 1)** -- check if completely done in zyBook __X__ ; otherwise, discuss issues below
Include a screenshot of current status/score

Exercise 4 – **zyBook 7.30 LAB: Vending Machine** -- check if completely done in zyBook __X__ ; otherwise, discuss issues below
Include a screenshot of current status/score

Exercise 5 – **Height Class Version 1** -- check if completely done __X__ ; otherwise, discuss issues below

Source code below:

Driver program:

```
/*
    Program: Height Class V1
    Author: Ali Mortada
    Class: CSCI 140
    Date: 4/6/23
    Description: Driver program for Height class. Stores height as feet
and inches in a class.
    Can also print stored height and adjust height by adding one inch.
    I certify that the code below is my own work.
    Exception(s): N/A
*/

#include <iostream>
#include "Height.h"
using namespace std;

int main() {

    cout << "Author: Ali Mortada" << endl << endl;

    // Create Height objects
    Height h3(5, 8);           // 5 feet, 8 inches, output: 5' 8"
    Height h4(-1, 5);          // 0 feet, 5 inches, output: 0' 5" (default
feet used)
    Height h5(6, 15);          // 6 feet, 0 inches, output: 6' 0" (default
inches used)

    // Print height h3
    cout << "h3: ";
    h3.print();                // h3: 5' 8"
    cout << endl;

    // Print height h4
    cout << "h4: ";
    h4.print();                // h4: 0' 5"
    cout << endl;

    // Print height h5
```

```
    cout << "h5: ";
    h5.print();                // h5: 6' 0"
    cout << endl;

    // Perform various operations
    h3.setFeet(-2);            // 5 feet, 8 inches (feet stay the same)
    h3.setInches(10);          // 5 feet, 10 inches
    cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() <<
endl;  // 5' 10"

    h4.setFeet(6);             // 6 feet, 5 inches
    h4.setInches(12);          // 6 feet, 5 inches (inches stay the same)
    cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() <<
endl;  // 6' 5"

    h5.setInches(10);          // 6 feet, 10 inches
    h5.adjust();               // 6 feet, 11 inches
    h5.adjust();               // 7 feet, 0 inches
    h5.print();                // 7' 0"

    cout << endl;

    return 0;
}
```

Class declaration:

```
#ifndef HEIGHT_H_
#define HEIGHT_H_

#include <iostream>
using namespace std;

class Height {
    public:
        Height(int userFeet, int userInches);
        void setFeet(int userFeet);
        void setInches(int userInches);
        int getFeet() const;
        int getInches() const;
        void print() const;
        void adjust();
    private:
        int feet;
        int inches;
```

```
};

#endif
```

Class definition:

```
#include <iostream>
#include "Height.h"
using namespace std;

// Default constructor
Height::Height(int userFeet, int userInches) {
    // Set feet to input if >= 0, if not then default to 0
    if (userFeet >= 0) {
        feet = userFeet;
    } else {
        feet = 0;
    }
    // Set inches to input if between 0 and 11, if not then default to 0
    if (userInches >= 0 && userInches <= 11) {
        inches = userInches;
    } else {
        inches = 0;
    }
}

// Mutator for feet data member
void Height::setFeet(int userFeet) {
    if (userFeet >= 0)
        feet = userFeet;
}

// Mutator for inches data member
void Height::setInches(int userInches) {
    if (userInches >= 0 && userInches <= 11)
        inches = userInches;
}

// Accessor for feet data member
int Height::getFeet() const {
    return feet;
}

// Accessor for inches data member
int Height::getInches() const {
```

```
    return inches;
}

// Print height in ' " format
void Height::print() const {
    cout << feet << "' " << inches << "\"" << endl;
}

// Increments inch by 1 and, if needed, feet by 1
void Height::adjust() {
    if (inches != 11) {
        inches++;
    } else {
        inches = 0;
        feet++;
    }
}
```

Input/output below:

```
PS C:\Users\Ali\Documents\Mt. SAC\CSCI 140\CSCI-140\PA 6> g++ HeightApp.cpp Height.cpp -o HeightApp
PS C:\Users\Ali\Documents\Mt. SAC\CSCI 140\CSCI-140\PA 6> ./HeightApp
Author: Ali Mortada
h3: 5' 8"

h4: 0' 5"

h5: 6' 0"

feet: 5, inches: 10
feet: 6, inches: 5
7' 0"
```

Answer for Question 1

Grouping multiple different data items of the same type in one struct makes your code more clear and can sometimes make it easier to access certain data values. The main difference between a struct and class in C++ is that by default, a struct's data members are public and can be easily accessed by the programmer, while in a class, data members are private by default and can only be accessed by member functions. A struct only allows variables to be grouped together, while a class can also contain member functions.

Answer for Question 2

A default constructor is a constructor that gives an object default values when created without any parameters. Overloading constructors makes it so that the object is still created, even if not all parameters are provided.

Extra Credit – provide if applicable.