

CSCI 140 PA 7 Submission

Due Date: 4/21/2025

Name(s): Ean Zheng

Exercise 1 – 7.25 LAB: Nutritional information (classes/constructors)

The screenshot displays the zyBooks lab environment for CSCI 140. The lab title is "7.25 LAB: Nutritional information (classes/constructors)". The instructions describe the task: to complete the `FoodItem` class with constructors to initialize food items. The default constructor initializes the name to "Water" and all data members to 0.0. The second constructor takes four parameters (food name, grams of fat, grams of carbohydrates, and grams of protein) and assigns each class data member with the appropriate parameter value.

Example input and output for "Water":

```
Water
the output is:
Nutritional information per serving of Water:
Fat: 0.00 g
Carbohydrates: 0.00 g
Protein: 0.00 g
Number of calories for 1.00 serving(s): 0.00
```

Example input and output for "M&M's":

```
M&M's
10.0
34.0
2.0
3.0
where M&M's is the food name, 10.0 is the grams of fat, 34.0 is the grams of carbohydrates, 2.0 is the grams of protein, and 3.0 is the number of servings, the output is:
Nutritional information per serving of M&M's:
Fat: 10.00 g
Carbohydrates: 34.00 g
Protein: 2.00 g
Number of calories for 1.00 serving(s): 234.00
Number of calories for 3.00 serving(s): 702.00
```

Note: The program outputs the number of calories for one serving of a food and for the input number of servings as well. The program only outputs the calories for one serving of water.

The code editor shows the implementation of the `FoodItem` class in `FoodItem.cpp` and `FoodItem.h`. The `FoodItem.h` file includes `FoodItem.h`, `iostream`, and `iomanip`. The `FoodItem` class is defined with four data members: `name`, `fat`, `carbs`, and `protein`. The `FoodItem` class has two constructors: a default constructor and a parameterized constructor. The `FoodItem` class also has four public methods: `getName()`, `getfat()`, `getcarbs()`, and `getprotein()`. The `FoodItem` class also has a public method `getcalories()` that calculates the number of calories for a given number of servings. The `main` function in `main.cpp` uses the `FoodItem` class to create two `FoodItem` objects, `water` and `mms`, and prints their nutritional information.

```
1 #include "FoodItem.h"
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 FoodItem::FoodItem(string n, double f, double c, double p) {
7     name = n;
8     fat = f;
9     carbs = c;
10    protein = p;
11 }
12
13 string FoodItem::getName() {
14     return name;
15 }
16
17 double FoodItem::getfat() {
18     return fat;
19 }
20
21 double FoodItem::getcarbs() {
22     return carbs;
23 }
24
25 double FoodItem::getprotein() {
26     return protein;
27 }
28
29 double FoodItem::getcalories(double numservings) {
30     // calorie formula
31     double calories = ((fat * 9) + (carbs * 4) + (protein * 4)) * number;
32     return calories;
33 }
34
35 void FoodItem::PrintInfo() {
36     cout << fixed << setprecision(2);
37     cout << "Nutritional information per serving of " << name << " is: " << endl;
38     cout << "Fat: " << fat << " g" << endl;
39     cout << "Carbohydrates: " << carbs << " g" << endl;
40     cout << "Protein: " << protein << " g" << endl;
41 }
42
```

Calendar MATH-181-09-4131... PA 7 Submission Section 7.25 - CSCI 140 C++ L... Section 8.1 - CSCI 140 C++ L... Demos | Graphing Calculator

learn.zybooks.com/zybook/MTSACCSCI140V0Spring2025/chapter7/section25

Google Chrome isn't your default browser Set as default

zyBooks My library - CSCI 140 C++ Language and Object Development home - 7.25 LAB: Nutritional Information (classes/constructors)

zybooks catalog Help/FAQ

Files

- FoodItem.cpp
- FoodItem.h
- main.cpp

```
1 #ifndef FOODITEM_H
2 #define FOODITEM_H
3
4 #include <string>
5 using namespace std;
6
7
8 class FoodItem {
9 public:
10     FoodItem(string n = "water", double f = 0.0, double c = 0.0, double p = 0.0) {}
11     string GetName();
12     double GetFat();
13     double GetCarbs();
14     double GetProtein();
15     double GetCalories(double numServings);
16     void PrintInfo();
17 private:
18     string name;
19     double fat;
20     double carbs;
21     double protein;
22 };
23 #endif
```

Submit for grading

Coding trial of your work [What is this?](#)

4/19 0:10 ML018

Latest submission - 12:59 PM PDT on 04/19/25 Submission passed all tests ✓ Total score: 10 / 10

☐ Only show failing tests [Open submissions code](#)

1:01 PM 4/19/2025

Exercise 2 – 7.26 LAB: Artwork label

The screenshot shows the zyBooks lab interface for '7.26 LAB: Artwork label (classes/constructors)'. The lab is titled '7.26 LAB: Artwork label (classes/constructors)' and is marked as 'Full screen' and '10 / 10'.

The lab instructions state: "Given main(), complete the `Artist` class (in files `Artist.h` and `Artist.cpp`) with constructors to initialize an artist's information, get member functions, and a `PrintInfo()` member function. The default constructor should initialize the artist's name to 'unknown' and the years of birth and death to -1. `PrintInfo()` displays 'Artist', then a space, then the artist's name, then another space, then the birth and death dates in one of three formats:

- (XXXX to YYYY) if both the birth and death years are nonnegative
- (XXXX to present) if the birth year is nonnegative and the death year is negative
- (unknown) otherwise

Complete the `Artwork` class (in files `Artwork.h` and `Artwork.cpp`) with constructors to initialize an artwork's information, get member functions, and a `PrintInfo()` member function. The default constructor should initialize the title to 'unknown', the year created to -1. `PrintInfo()` displays an artist's information by calling the `PrintInfo()` function in the `Artist` class, followed by the artwork's title and the year created. Declare a private field of type `Artist` in the `Artwork` class.

Ex. If the input is:

```
Pablo Picasso
Three Musicians
1881
1973
1921
```

1881 and 1973 being the birth and death years respectively, with 1921 being the year the work was created, the output is:

```
Artist: Pablo Picasso (1881 to 1973)
Title: Three Musicians, 1921
```

Ex. If the input is:

```
Brice Marden
Distant Muses
1938
-1
2000
```

the output is:

```
Artist: Brice Marden (1938 to present)
Title: Distant Muses, 2000
```

The code editor shows the following code:

```
1 #include "Artwork.h"
2 #include <iostream>
3
4 Artwork::Artwork(){
5     title = "unknown";
6     yearCreated = -1;
7 }
8
9 Artwork::Artwork(string title, int yearCreated, Artist artist){
10     this->title = title;
11     this->yearCreated = yearCreated;
12     this->artist = artist;
13 }
14
15 string Artwork::getTitle(){
16     return title;
17 }
18
19 int Artwork::getYearCreated(){
20     return yearCreated;
21 }
22
23 void Artwork::PrintInfo(){
24     artist.PrintInfo();
25     cout << "Title: " << title << ", " << yearCreated << endl;
26 }
```

The console output shows:

```
Artist: Pablo Picasso (1881 to 1973)
Title: Three Musicians, 1921
```

The interface includes a 'Submit for grading' button and a 'Run' button.

cs140PASubmissionV1.docx C: x x PA 7 Submission x Section 7.26 - CSCI 140 C++ L: x Section 8.1 - CSCI 140 C++ L: x Demos | Graphing Calculator x +

learn.zybooks.com/zybook/MTSACCS140V1Spring2025/chapter7/section26

Google Chrome isn't your default browser Set as default

zyBooks My library > CSCI 140 C++ Language and Object Development home > 7.26: LAB: Artwork label (classes/constructors) zyBooks catalog Help/FAQ Ean Zheng

Open new tab Dock

Run

Artwork x Artwork.cpp x Artist x Artist.cpp x main.cpp x

```
1 #ifndef ARTWORKH
2 #define ARTWORKH
3
4 #include "Artist.h"
5
6 class Artwork{
7 public:
8     Artwork();
9
10    Artwork(string title, int yearCreated, Artist artist);
11
12    string GetTitle();
13
14    int GetYearCreated();
15
16    void PrintInfo();
17
18 private:
19    string title;
20    int yearCreated;
21    Artist artist;
22
23 };
24
25 #endif
```

DESKTOP CONSOLE

Submit for grading

cs140PASubmissionV1.docx C: x x PA 7 Submission x Section 7.26 - CSCI 140 C++ L: x Section 8.1 - CSCI 140 C++ L: x Demos | Graphing Calculator x +

learn.zybooks.com/zybook/MTSACCS140V1Spring2025/chapter7/section26

Google Chrome isn't your default browser Set as default

zyBooks My library > CSCI 140 C++ Language and Object Development home > 7.26: LAB: Artwork label (classes/constructors) zyBooks catalog Help/FAQ Ean Zheng

Run

Artwork x Artwork.cpp x Artist x Artist.cpp x main.cpp x

```
1 #include "Artist.h"
2 #include <iostream>
3 #include <string>
4 using namespace std;
5
6 Artist::Artist(){
7     artistName = "unknown";
8     birthYear = -1;
9     deathYear = -1;
10 }
11
12 Artist::Artist(string artistName, int birthYear, int deathYear){
13     this->artistName = artistName;
14     this->birthYear = birthYear;
15     this->deathYear = deathYear;
16 }
17
18 string Artist::getArtistName() const{
19     return artistName;
20 }
21
22 int Artist::getBirthYear() const{
23     return birthYear;
24 }
25
26 int Artist::getDeathYear() const{
27     return deathYear;
28 }
29
30 void Artist::PrintInfo() const{
31     cout << "Artist: " << artistName << " ";
32     if(birthYear >= 0 && deathYear >= 0){
33         cout << "[" << birthYear << " to " << deathYear << "]";
34     } else if(birthYear >= 0 && deathYear < 0){
35         cout << "[" << birthYear << " to present)";
36     } else{
37         cout << "(unknown)";
38     }
39 }
```

DESKTOP CONSOLE

Submit for grading

cs140PASubmission01.docx C: x x PA 7 Submission x Section 7.26 - CSCI 140 C++ L: x Section 8.1 - CSCI 140 C++ L: x Demos | Graphing Calculator x +

learn.zybooks.com/zybook/MTSACCSCI140VnSpring2025/chapter7/section26

Google Chrome isn't your default browser [Set as default](#)

zyBooks My library > CSCI 140 C++ Language and Object Development home > 7.26: LAB: Artwork label (classes/constructors) zyBooks catalog Help/FAQ Ean Zheng

Run

Artwork.h x Artwork.cpp x Artist.h x Artist.cpp x main.cpp x

```
1 #ifndef ARTISTA
2 #define ARTISTA
3
4 #include <string>
5 using namespace std;
6
7 class Artist{
8 public:
9     Artist();
10    Artist(string artistname, int birthyear, int deathyear);
11    string getname() const;
12    int getbirthyear() const;
13    int getdeathyear() const;
14    void PrintInfo() const;
15
16 private:
17    string artistname;
18    int birthyear;
19    int deathyear;
20 }
21
22 #endif
```

DESKTOP CODESHELL

Submit for grading

Coding trail of your work [What is this?](#)

4/19 8:0, 0, 9, 0, 9, 0, 9, 0, 9, 10 mins:27

Latest submission - 2:09 PM PDT on 04/19/25 Submission passed all tests ✓ Total score: 10 / 10

☐ Only show failing tests [Open submission's code](#)

1. Console output 1/1

2:18 PM 4/19/2025

Exercise 3 – 7.28 LAB*: Warm up: Online shopping cart (Part 2) – more points for this exercise

The screenshot displays the zyBooks online shopping cart lab interface. The top section shows the lab title "7.28.1: LAB* Program: Online shopping cart (Part 2)" and a brief description. Below this, the instructions for Step 1 and Step 2 are provided.

Step 1 (3 pts): Extend the ItemToPurchase class per the following specifications:

- Parameterized constructor to assign item name, item description, item price, and item quantity (default values of 0, 1 pt)
- Public member functions
 - setDescription() mutator & getDescription() accessor (2 pts)
 - PrintItemCost() - Outputs the item name followed by the quantity, price, and subtotal
 - PrintItemDescription() - Outputs the item name and description
- Private data members
 - string itemDescription - initialized in default constructor to "none"

Ex. of PrintItemCost() output:
Bottled Water 10 @ \$1 = \$10

Ex. of PrintItemDescription() output:
Bottled Water: Deer Park, 12 oz.

Step 2 (7 pts): Build three new files:

- ShoppingCart.h - Class declaration
- ShoppingCart.cpp - Class definition
- main.cpp - main() function (Note: main() functionality differs from the previous program)

Build the ShoppingCart class with the following specifications.

- Default constructor
- Parameterized constructor which takes the customer name and date as parameters (1 pt)
- Private data members
 - string customerName - initialized in default constructor to "none"
 - string currentDate - initialized in default constructor to "January 1, 2014"
 - vector<ItemToPurchase> cartItems
- Public member functions
 - GetCustomerName() accessor (1 pt)
 - GetDate() accessor (1 pt)
 - AddItem() - Adds an item to cartItems vector. Has parameter ItemToPurchase. Does not return anything.
 - RemoveItem() - Removes item from cartItems vector. Has a string (an item's name) parameter. Does not return anything.
 - If item name cannot be found, output a message: Item not found in cart. Nothing removed.
 - ModifyItem() - Modifies an item's description, price, and/or quantity. Has parameter ItemToPurchase. Does not return anything.
 - If item can be found, the user's input is used; if not, check if parameter has default values for description, price, and quantity. If not, nothing is changed.

The bottom section shows the code for the ShoppingCart class in main.cpp. The code includes the necessary headers and implements the main function, which uses the ShoppingCart class to manage a shopping cart.

```
main.cpp | ShoppingCart.h | ItemToPurchase.cpp | ShoppingCart.h | ItemToPurchase.h
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 #include "ShoppingCart.h"
6
7 void PrintMenu() {
8     cout << "Menu" << endl;
9     cout << "a - add item to cart" << endl;
10    cout << "r - remove item from cart" << endl;
11    cout << "c - change item quantity" << endl;
12    cout << "d - output item's description" << endl;
13    cout << "p - output shopping cart" << endl;
14    cout << "q - quit" << endl;
15 }
16
17 void executeMenu(char option, ShoppingCart theCart) {
18     if (option == 'a') {
19         cout << "Add item to cart" << endl;
20         string name;
21         string description;
22         int price;
23         int quantity;
24         cout << "Enter the item name:" << endl;
25         getline(cin, name);
26         cout << "Enter the item description:" << endl;
27         getline(cin, description);
28         cout << "Enter the item price:" << endl;
29         cin >> price;
30         cout << "Enter the item quantity:" << endl;
31         cin >> quantity;
32         theCart.addItem(name, description, price, quantity);
33     }
34     else if (option == 'r') {
35         cout << "Remove item from cart" << endl;
36         string name;
37         cout << "Enter name of item to remove:" << endl;
38         getline(cin, name);
39         theCart.removeItem(name);
40     }
41     else if (option == 'c') {
42         cout << "Change item quantity" << endl;
43         string name;
44         int quantity;
45         cout << "Enter the item name:" << endl;
46         getline(cin, name);
47         cout << "Enter the new quantity:" << endl;
48         cin >> quantity;
49     }
50 }
```

cs140PASubmission1do... x | wif PA 7 Submission x | My activity - CSQ 140 C... x | Section 7.17 - CSQ 140... x | Section 7.27 - CSQ 140... x | Section 7.28 - CSQ 140... x | Section 8.1 - CSQ 140 C... x | c++ on input not worki... x | + -

learn.zybooks.com/zybook/MTSACCSQ140V6Spring2025/chapter/7/section/28/content_resource_id=108377958

Google Chrome isn't your default browser Set as default

zyBooks My library > CSQ 140 C++ Language and Object Development home > 7.28 LAB* Program: Online shopping cart (Part 2) zyBooks catalog Help/FAQ Ean Zheng

main.cpp x ShoppingCart.cpp x ItemToPurchase.cpp x ShoppingCart.h x ItemToPurchase.h x History Tutorial

```
main.cpp
41 thecart.removeItem(name);
42 else if (option == "C") {
43     cout << "REMOVE ITEM QUANTITY" << endl;
44     string name;
45     int quantity;
46     cout << "Enter the item name:" << endl;
47     cin.ignore();
48     getline(cin, name);
49     cout << "Enter the new quantity:" << endl;
50     cin >> quantity;
51     thecart.modifyItem(changingItem, name, quantity);
52     thecart.modifyItem(changingItem);
53 }
54 else if (option == "I") {
55     cout << "OUTPUT ITEM'S DESCRIPTION" << endl;
56     thecart.printDescription();
57 }
58 else if (option == "S") {
59     cout << "SUM OF SHOPPING CART" << endl;
60     thecart.printTotal();
61 }
62 }
63
64 int main() {
65     string name;
66     string date;
67     char choice;
68     cout << "Enter customer's name:" << endl;
69     getline(cin, name);
70     cout << "Enter today's date:" << endl;
71     getline(cin, date);
72     cout << endl;
73     ShoppingCart cart(name, date);
74     cout << "Customer name: " << name << endl;
75     cout << "Today's date: " << date << endl;
76     cout << endl;
77     cout << "Choose an option:" << endl;
78     while (choice != "q") {
79         char choice;
80         cout << endl;
81         printMenu();
82         cout << endl;
83         cout << "Choose an option:" << endl;
84         cin >> choice;
85     }
86     return 0;
87 }
88 }
```

cs140PASubmission1do... x | wif PA 7 Submission x | My activity - CSQ 140 C... x | Section 7.17 - CSQ 140... x | Section 7.27 - CSQ 140... x | Section 7.28 - CSQ 140... x | Section 8.1 - CSQ 140 C... x | c++ on input not worki... x | + -

learn.zybooks.com/zybook/MTSACCSQ140V6Spring2025/chapter/7/section/28/content_resource_id=108377958

Google Chrome isn't your default browser Set as default

zyBooks My library > CSQ 140 C++ Language and Object Development home > 7.28 LAB* Program: Online shopping cart (Part 2) zyBooks catalog Help/FAQ Ean Zheng

main.cpp x ShoppingCart.cpp x ItemToPurchase.cpp x ShoppingCart.h x ItemToPurchase.h x History Tutorial

```
main.cpp
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 #include "ShoppingCart.h"
5
6 ShoppingCart::ShoppingCart() {
7     customerName = "none";
8     currentDate = "January 1, 2025";
9 }
10
11 ShoppingCart::ShoppingCart(string name, string date) {
12     customerName = name;
13     currentDate = date;
14 }
15
16 string ShoppingCart::getCustomerName() const {
17     return customerName;
18 }
19
20 string ShoppingCart::getDate() const {
21     return currentDate;
22 }
23
24 void ShoppingCart::addItem(ItemToPurchase item) {
25     cartItems.push_back(item);
26 }
27
28 void ShoppingCart::removeItem(string name) {
29     for (int i = 0; i < cartItems.size(); ++i) {
30         if (cartItems.at(i).getName() == name) {
31             cartItems.erase(cartItems.begin() + i);
32             return;
33         }
34     }
35     cout << "Item not found in cart. Nothing removed." << endl;
36 }
37
38 void ShoppingCart::modifyItem(ItemToPurchase item) {
39     for (int i = 0; i < cartItems.size(); ++i) {
40         if (cartItems.at(i).getName() == item.getName()) {
41             if (item.getDescription() != "none") {
42                 cartItems.at(i).setDescription(item.getDescription());
43             }
44             if (item.getPrice() != 0) {
45                 cartItems.at(i).setPrice(item.getPrice());
46             }
47             if (item.getQuantity() != 0) {
48                 cartItems.at(i).setQuantity(item.getQuantity());
49             }
50         }
51     }
52     cout << "Item not found in cart. Nothing modified." << endl;
53 }
54
55 int ShoppingCart::getNumberOfCart() const {
56     int totalNumber = 0;
57     for (int i = 0; i < cartItems.size(); ++i) {
58         totalNumber += cartItems.at(i).getQuantity();
59     }
60 }
```

cs140FASubmissionV1... PA 7 Submission My activity - CSI 140 C... Section 7.17 - CSI 140... Section 7.27 - CSI 140... Section 7.28 - CSI 140... Section 8.1 - CSI 140 C... c++ cin input not worki...
learn.zybooks.com/zybook/MTSACCS140NoSpring2025/chapter/7/section/28/content_resource_id=108377958
Google Chrome isn't your default browser Set as default
zyBooks My library - CSI 140 C++ Language and Object Development home - 7.28 LAB* Program: Online shopping cart (Part 2) zyBooks catalog Help/FAQ Ean Zheng

```
main.cpp | ShoppingCart.cpp | ItemToPurchase.cpp | ShoppingCart.h | ItemToPurchase.h
44
45 }
46 }
47 cout << "Item not found in cart. Nothing modified." << endl;
48 }
49 int ShoppingCart::getNumItems() const {
50     int totalNumber = 0;
51     for (int i = 0; i < cartItems.size(); ++i) {
52         totalNumber += cartItems.at(i).getQuantity();
53     }
54     return totalNumber;
55 }
56 int ShoppingCart::getCostOfCart() {
57     int totalCost = 0;
58     for (int i = 0; i < cartItems.size(); ++i) {
59         totalCost += cartItems.at(i).getPrice() * cartItems.at(i).getQuantity();
60     }
61     return totalCost;
62 }
63 void ShoppingCart::printTotal() {
64     cout << "Customer Name: " << this->customerName << " - " << currentDate << endl;
65     int totalNumber = 0;
66     for (int i = 0; i < cartItems.size(); ++i) {
67         totalNumber += cartItems.at(i).getQuantity();
68     }
69     cout << "Number of Items: " << totalNumber << endl << endl;
70     int totalCost = 0;
71     if (cartItems.empty()) {
72         cout << "Shopping cart is empty" << endl;
73     }
74     else {
75         for (int i = 0; i < cartItems.size(); ++i) {
76             cartItems.at(i).printItemCost();
77             totalCost += cartItems.at(i).getPrice() * cartItems.at(i).getQuantity();
78         }
79         cout << endl;
80         cout << "Total: $" << totalCost << endl;
81     }
82 void ShoppingCart::printDescriptions() {
83     cout << "Customer Name: " << this->customerName << " - " << currentDate << endl << endl;
84     cout << "Item Descriptions" << endl;
85     if (cartItems.empty()) {
86         cout << "Shopping cart is empty" << endl;
87     }
88     else {
89         for (int i = 0; i < cartItems.size(); ++i) {
90             cartItems.at(i).printItemDescription();
91         }
92     }
93 }
```

cs140FASubmissionV1... PA 7 Submission My activity - CSI 140 C... Section 7.17 - CSI 140... Section 7.27 - CSI 140... Section 7.28 - CSI 140... Section 8.1 - CSI 140 C... c++ cin input not worki...
learn.zybooks.com/zybook/MTSACCS140NoSpring2025/chapter/7/section/28/content_resource_id=108377958
Google Chrome isn't your default browser Set as default
zyBooks My library - CSI 140 C++ Language and Object Development home - 7.28 LAB* Program: Online shopping cart (Part 2) zyBooks catalog Help/FAQ Ean Zheng

```
main.cpp | ShoppingCart.cpp | ItemToPurchase.cpp | ShoppingCart.h | ItemToPurchase.h
1 #ifndef SHOPPING_CART_H
2 #define SHOPPING_CART_H
3
4 #include "ItemToPurchase.h"
5
6 #include <string>
7 #include <vector>
8 using namespace std;
9
10 class ShoppingCart {
11 public:
12     ShoppingCart();
13     ShoppingCart(string name, string date);
14     string getCustomerName() const;
15     string getDate() const;
16     void addItemToPurchase(Item);
17     void removeItem(string name);
18     void modifyItem(ItemToPurchase item);
19     int getNumItems() const;
20     int getCostOfCart();
21     void printTotal();
22     void printDescriptions();
23 private:
24     string customerName;
25     string currentDate;
26     vector<ItemToPurchase> cartItems;
27 };
28 #endif
```


cs1407ASubmissionV1... RA.7 Submission My activity - CSCI 140 C... Section 7.17 - CSCI 140 C... Section 7.27 - CSCI 140 C... Section 7.28 - CSCI 140 C... Section 8.1 - CSCI 140 C... c++ in input net workin... + -

learn.zybooks.com/zybook/MTSACCS140/spring2025/chapter/7/section/28/content_resource_id=108377958

Google Chrome isn't your default browser Set as default

zyBooks My library - CSCI 140 C++ Language and Object Development home - 7.28 LAB* Program: Online shopping cart (Part 2) zyBooks catalog Help/FAQ Ean Zheng

main.cpp x ShoppingCart.cpp x ItemToPurchase.cpp x ShoppingCart.h x ItemToPurchase.h x

```
1 #include <iostream>
2 using namespace std;
3
4 #include "ItemToPurchase.h"
5
6 ItemToPurchase::ItemToPurchase() {
7     itemName = "none";
8     itemPrice = 0;
9     itemQuantity = 0;
10 }
11
12 ItemToPurchase::ItemToPurchase(string name, string description, int price, int quantity) {
13     itemName = name;
14     itemDescription = description;
15     itemPrice = price;
16     itemQuantity = quantity;
17 }
18
19 void ItemToPurchase::setName(string name) {
20     itemName = name;
21 }
22 string ItemToPurchase::getName() const {
23     return itemName;
24 }
25 void ItemToPurchase::setDescription(string description) {
26     itemDescription = description;
27 }
28 string ItemToPurchase::getDescription() const {
29     return itemDescription;
30 }
31 void ItemToPurchase::setPrice(int price) {
32     itemPrice = price;
33 }
34 int ItemToPurchase::getPrice() const {
35     return itemPrice;
36 }
37 void ItemToPurchase::setQuantity(int quantity) {
38     itemQuantity = quantity;
39 }
40 int ItemToPurchase::getQuantity() const {
41     return itemQuantity;
42 }
43 void ItemToPurchase::printItemCost() {
44     cout << itemName << " " << itemQuantity << " @ $" << itemPrice << " = $" << itemQuantity * itemPrice << endl;
45 }
46 void ItemToPurchase::printItemDescription() {
47     cout << itemName << ": " << itemDescription << endl;
48 }
```

DESKTOP CONSOLE

cs1407ASubmissionV1... RA.7 Submission My activity - CSCI 140 C... Section 7.17 - CSCI 140 C... Section 7.27 - CSCI 140 C... Section 7.28 - CSCI 140 C... Section 8.1 - CSCI 140 C... c++ in input net workin... + -

learn.zybooks.com/zybook/MTSACCS140/spring2025/chapter/7/section/28/content_resource_id=108377958

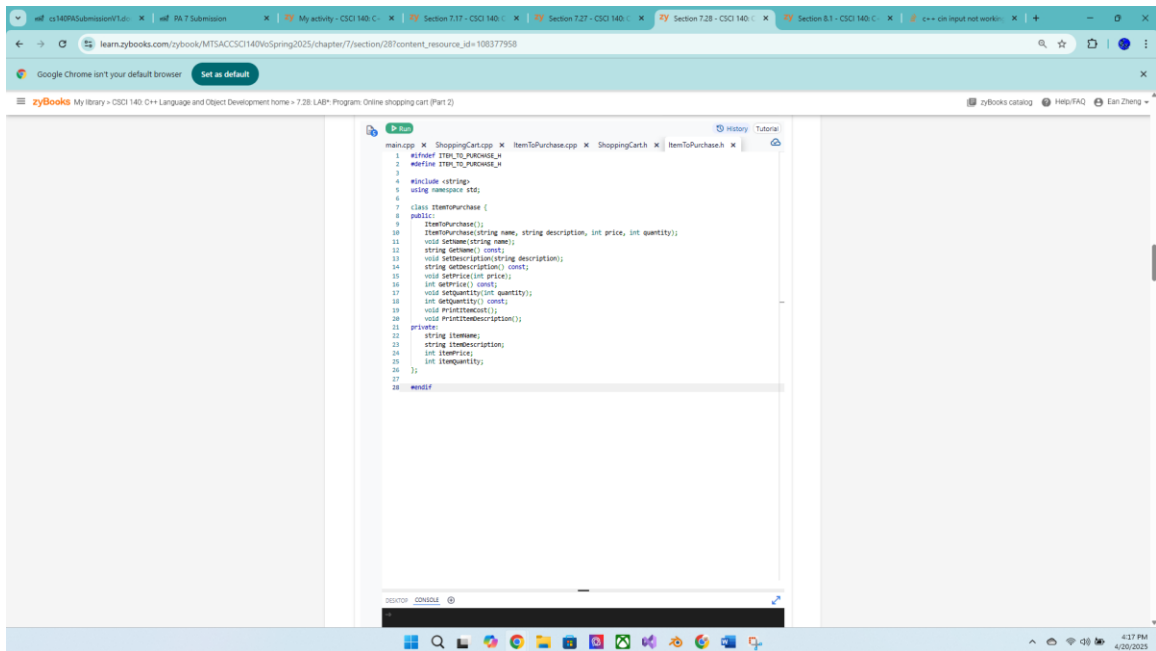
Google Chrome isn't your default browser Set as default

zyBooks My library - CSCI 140 C++ Language and Object Development home - 7.28 LAB* Program: Online shopping cart (Part 2) zyBooks catalog Help/FAQ Ean Zheng

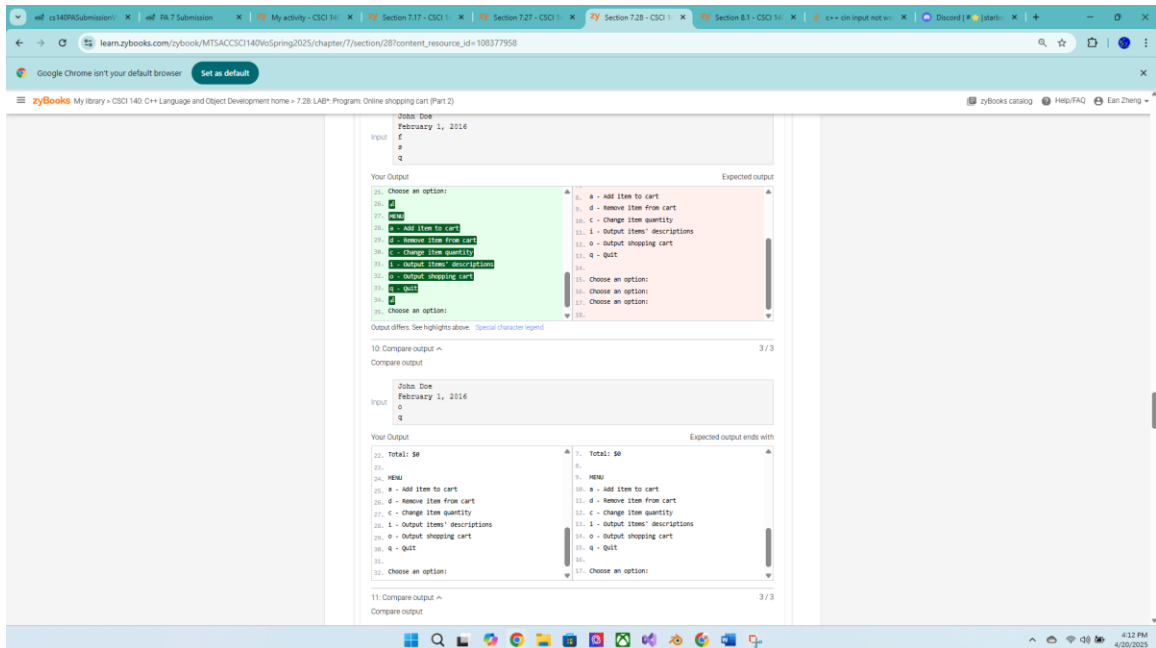
main.cpp x ShoppingCart.cpp x ItemToPurchase.cpp x ShoppingCart.h x ItemToPurchase.h x

```
1 #include <iostream>
2 using namespace std;
3
4 #include "ItemToPurchase.h"
5
6 ItemToPurchase::ItemToPurchase() {
7     itemName = "none";
8     itemPrice = 0;
9     itemQuantity = 0;
10 }
11
12 ItemToPurchase::ItemToPurchase(string name, string description, int price, int quantity) {
13     itemName = name;
14     itemDescription = description;
15     itemPrice = price;
16     itemQuantity = quantity;
17 }
18
19 void ItemToPurchase::setName(string name) {
20     itemName = name;
21 }
22 string ItemToPurchase::getName() const {
23     return itemName;
24 }
25 void ItemToPurchase::setDescription(string description) {
26     itemDescription = description;
27 }
28 string ItemToPurchase::getDescription() const {
29     return itemDescription;
30 }
31 void ItemToPurchase::setPrice(int price) {
32     itemPrice = price;
33 }
34 int ItemToPurchase::getPrice() const {
35     return itemPrice;
36 }
37 void ItemToPurchase::setQuantity(int quantity) {
38     itemQuantity = quantity;
39 }
40 int ItemToPurchase::getQuantity() const {
41     return itemQuantity;
42 }
43 void ItemToPurchase::printItemCost() {
44     itemName << " " << itemQuantity << " @ $" << itemPrice << " = $" << itemQuantity * itemPrice << endl;
45 }
46 void ItemToPurchase::printItemDescription() {
47     itemName << ": " << itemDescription << endl;
48 }
```

DESKTOP CONSOLE



1 Wrong because 1 test case isn't satisfied from menu printing each choice loop iteration:



Many wrong because each choice loop iteration doesn't print menu:

The image displays two screenshots of a zyBooks online shopping cart program. The top screenshot shows a comparison of the user's output with the expected output for a shopping cart. The bottom screenshot shows a comparison of the user's output with the expected output for a shopping cart menu.

Top Screenshot:

11: Compare output

Compare output

John Doe
February 1, 2016
a
Blue Romulose
Vuit doine, Weightlifting shoes
169
2
o
q

Your Output

```
14. Choose an option:
15. Enter shopping cart:
16. John Doe's Shopping Cart - February 1, 2016
17. Number of items: 6
18. q
19. Shopping cart is empty
20. Total: $0
21. Total: $0
22. Choose an option:
23. Choose an option:
```

Output is nearly correct, but whitespace differs. See highlights above. [Special character legend](#)

Expected output ends with

```
14. q - Quit
15. q - Quit
```

12: Compare output

Compare output

Total: \$376

Your Output

```
28. Total: $376
29. Choose an option:
30. Choose an option:
```

Output differs. See highlights above. [Special character legend](#)

Expected output ends with

```
28. Total: $376
29. Add item to cart
30. Remove item from cart
31. Change item quantity
32. Output items' descriptions
33. Output shopping cart
34. q - Quit
35. Choose an option:
```

Bottom Screenshot:

10: Compare output

Compare output

John Doe
February 1, 2016
a
q

Your Output

```
14. Choose an option:
15. Enter shopping cart:
16. John Doe's Shopping Cart - February 1, 2016
17. Number of items: 6
18. q
19. Shopping cart is empty
20. Total: $0
21. Total: $0
22. Choose an option:
23. Choose an option:
```

Output is nearly correct, but whitespace differs. See highlights above. [Special character legend](#)

Expected output

```
1. Add item to cart
2. Remove item from cart
3. Change item quantity
4. Output items' descriptions
5. Output shopping cart
6. q - Quit
7. Choose an option:
8. Choose an option:
9. Choose an option:
```

11: Compare output

Compare output

John Doe
February 1, 2016
a
q

Your Output

```
14. Choose an option:
15. Enter shopping cart:
16. John Doe's Shopping Cart - February 1, 2016
17. Number of items: 6
18. q
19. Shopping cart is empty
20. Total: $0
21. Total: $0
22. Choose an option:
23. Choose an option:
```

Output is nearly correct, but whitespace differs. See highlights above. [Special character legend](#)

Expected output ends with

```
1. Enter shopping cart
2. John Doe's Shopping Cart - February 1, 2016
3. Number of items: 6
4. q
5. Shopping cart is empty
6. Total: $0
7. MENU
8. Add item to cart
9. Remove item from cart
```

Exercise 4 – Height class version 2 – more points for this exercise

Add the following operations to the Height class from previous lab and include additional test cases to add these three new features:

- Overload advance function so now there are two advance functions
 - o void increment(int inches);
 - o // make sure inches must be between 1 and 11
 - o // ignore invalid inches
- Returning the total inches.
 - o int totalInches() const;
- Overload operator == to compare the two heights. Can use either a member function or a friend function
 - o bool operator==(const Height &r) const; // member
 - o friend bool operator==(const Height &r, const Height &r);

Add the following test cases in your driver and add code to label height and new line for formatting as needed:

```
Height h6(0, 5); // feet: 0, inches: 5
// Print height (add code to label height and new line as needed)
h6.print(); // h6: 0' 5"
t6.increment(12); // ignore, feet: 0, inches: 5
t6.increment(-3); // ignore, feet: 0, inches: 5
t6.increment(); // feet: 0, inches: 6
t6.increment(10); // feet: 1, inches: 4
cout << "Total inches: " << h6.totalInches() << endl; // 16
// Print height (add code to label height and new line as needed)
h6.print(); // h6: 1' 4"
if (h3 == h6)
    cout << "h3 is the same as h6" << endl;
else
    cout << "h3 is not the same as h6" << endl;
// should output: h3 is not the same as h6
```

Source code below:

HeightV2.cpp:

/* Program: Height Class Version 2

Author: Ean Zheng

Class : CSCI 140

Date : 4/20/2025

Description :

I certify that the code below is my own work.

Exception(s) : N/A

*/

```

#include <iostream>
using namespace std;

#include "Height.h"

int main() {
    // Create 2 Height objects
    Height h1; // feet: 0, inches: 0
    Height h2(5); // feet: 5, inches: 0
    // Create some Height objects (same as original version)
    Height h3(5, 8); // feet: 5, inches: 8
    Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)
    Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)
    // Print height h3
    Height array[] { h1, h2, h3, h4, h5 };
    for (int i = 0; i < 5; ++i) {
        array[i].print();
    }
    cout << "h3: ";
    h3.print(); // h3: 5' 8"
    cout << endl;
    // Add more code below to print h4 and h5 like h3 above
    // Perform various operations
    h3.setFeet(-2); // feet: 5, inches: 8, feet stay the same
    h3.setInches(10); // feet: 5, inches: 10
    cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() << endl; // 5 10
    h4.setFeet(6); // feet: 6, inches: 5
    h4.setInches(12); // feet: 6, inches: 5, inches stay the same
    cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() << endl; // 6 5
    h5.setInches(10); // feet: 6, inches: 10
    h5.increment(); // feet: 6, inches: 11
    h5.increment(); // feet: 7, inches: 0
    cout << "h5: ";
    h5.print(); // h5: 7' 0"
    cout << endl;
    // Add more test cases if needed
    Height h6(0, 5); // feet: 0, inches: 5
    // Print height (add code to label height and new line as needed)
    cout << "h6: ";
    h6.print(); // h6: 0' 5"
    h6.increment(12); // ignore, feet: 0, inches: 5
    h6.increment(-3); // ignore, feet: 0, inches: 5
    h6.increment(); // feet: 0, inches: 6
    h6.increment(10); // feet: 1, inches: 4
    cout << "Total inches : " << h6.totalInches() << endl; // 16
    // Print height (add code to label height and new line as needed)

```

```

    cout << "h6: ";
    h6.print(); // h6: 1' 4"
    if (h3 == h6)
        cout << "h3 is the same as h6" << endl;
    else
        cout << "h3 is not the same as h6" << endl;
    // should output: h3 is not the same as h6
}

```

Height.cpp:

```

/* Program: Height Class Version 2
   Author: Ean Zheng
   Class : CSCI 140
   Date : 4/20/2025
   Description :
   I certify that the code below is my own work.
   Exception(s) : N/A
*/
#include <iostream>
using namespace std;

#include "Height.h"

Height::Height(int f, int i) : feet(f), inches(i) {
    if (feet < 0) feet = 0;
    if (inches < 0 || inches > 11) inches = 0;
}

void Height::setFeet(int f) {
    if (f >= 0) feet = f;
}

void Height::setInches(int i) {
    if (i >= 0 && i <= 11) inches = i;
}

int Height::getFeet() const {
    return feet;
}

int Height::getInches() const {
    return inches;
}

void Height::print() const {
    cout << feet << "' " << inches << "\" " << endl;
}

```

```

void Height::increment() {
    ++inches;
    if (inches == 12) {
        inches = 0;
        ++feet;
    }
}

void Height::increment(int inches) {
    if (inches > 0 && inches < 12) {
        this->inches = this->inches + inches;
        if (this->inches >= 12) {
            this->inches -= 12;
            ++feet;
        }
    }
}

int Height::totalInches() const {
    return (feet * 12) + inches;
}

bool Height::operator==(const Height& r) const {
    if (feet == r.getFeet() && inches == r.getInches())
        return true;
    else if (feet != r.getFeet() || inches != r.getInches())
        return false;
}

```

Height.h:

```

/* Program: Height Class Version 2
   Author: Ean Zheng
   Class : CSCI 140
   Date : 4/20/2025
   Description :
   I certify that the code below is my own work.
   Exception(s) : N/A
*/

#ifndef HEIGHT_H
#define HEIGHT_H

#include <string>
using namespace std;

class Height {

```

```

private:
    int feet;
    int inches;
public:
    Height(int f = 0, int i = 0);
    void setFeet(int f = 0);
    void setInches(int i = 0);
    int getFeet() const;
    int getInches() const;
    void print() const;
    void increment();
    void increment(int inches);
    int totalInches() const;
    bool operator==(const Height& r) const;
};

```

```

#endif

```

```

// Program: Height Class Version 2
// Author: Ean Zhang
// Class: CSC1100
// Date: 6/26/2020
// Description:
// I certify that the code below is my own work.
// Exception(s): N/A

// Include ciostream
using namespace std;

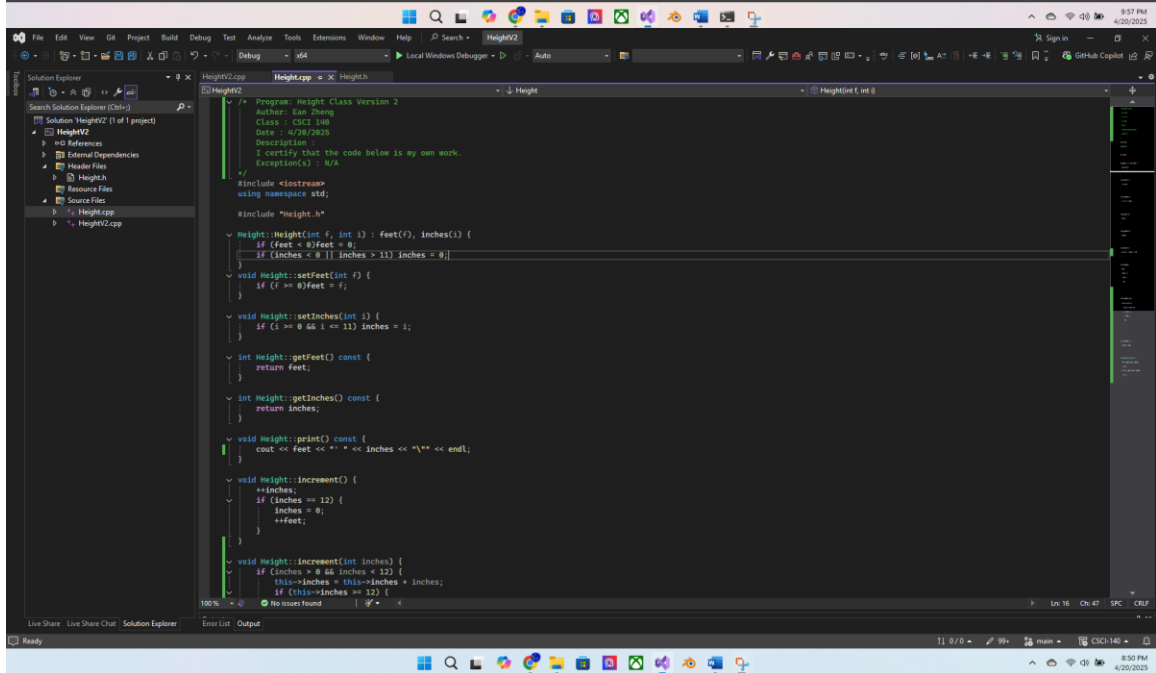
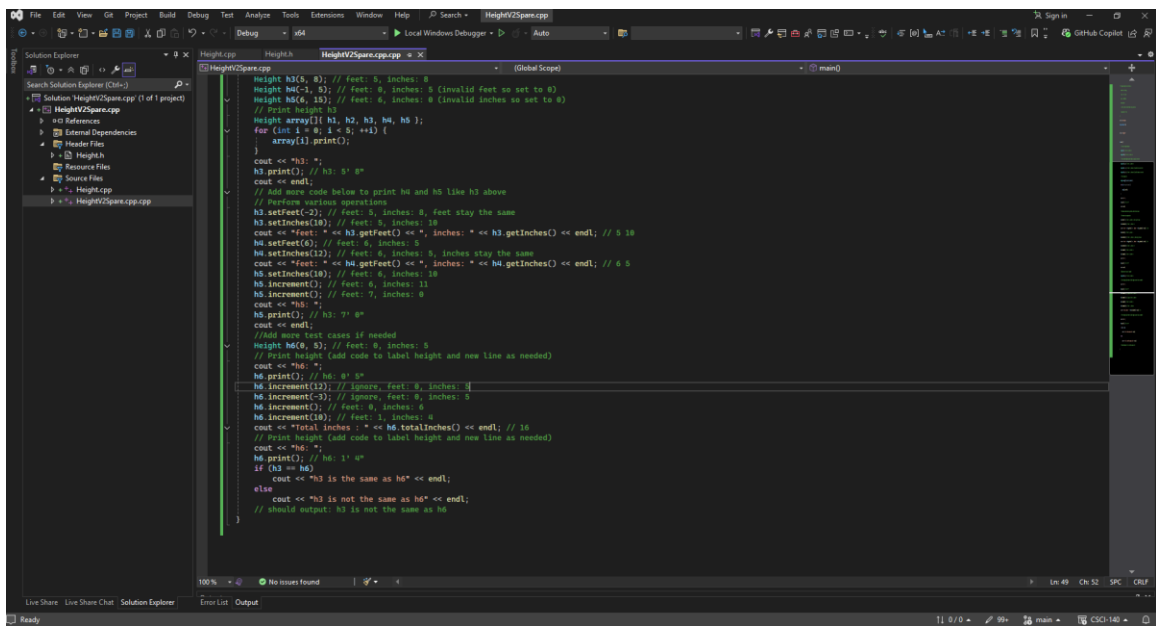
#include "Height.h"

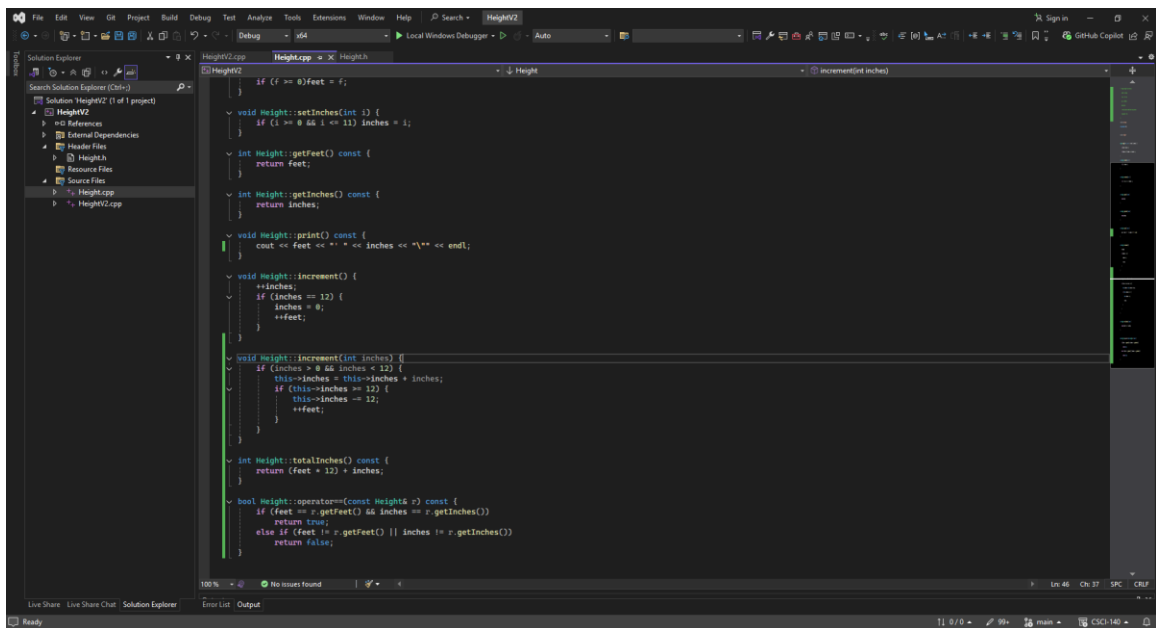
int main() {
    // Create 2 Height objects
    Height h1; // feet: 0, inches: 0
    Height h2(5); // feet: 5, inches: 0
    // Create some Height objects (same as original version)
    Height h3(5, 8); // feet: 5, inches: 8
    Height h4(-5, 8); // feet: 0, inches: 8 (invalid feet so set to 0)
    Height h5(5, 15); // feet: 5, inches: 0 (invalid inches so set to 0)
    // Print height h3
    Height array[1]; h1, h2, h3, h4, h5;
    for (int i = 0; i < 5; ++i) {
        array[i].print();
    }
    cout << "h3: ";
    h3.print(); // h3: 5' 8"
    cout << endl;

    // Add more code below to print h4 and h5 like h3 above
    // Perform various operations
    h3.setFeet(-2); // feet: 0, inches: 8, feet stay the same
    h3.setInches(18); // feet: 5, inches: 18
    cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() << endl; // 5 18
    h4.setFeet(6); // feet: 6, inches: 5
    h4.setInches(12); // feet: 6, inches: 5, inches stay the same
    cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() << endl; // 6 5
    h5.setInches(18); // feet: 5, inches: 18
    h5.increment(); // feet: 6, inches: 19
    h5.increment(); // feet: 7, inches: 19
    cout << "h5: ";
    h5.print(); // h5: 7' 19"
    cout << endl;

    // Add more test cases if needed
    Height h6(0, 5); // feet: 0, inches: 5
    // Print height (add code to label height and new line as needed)
    cout << "h6: ";
    h6.print(); // h6: 0' 5"
    h6.increment(12); // ignore, feet: 0, inches: 5
}

```



```
if (f == 0) feet = f;

void Height::setInches(int i) {
    if (i == 0 && i <= 11) inches = i;
}

int Height::getFeet() const {
    return feet;
}

int Height::getInches() const {
    return inches;
}

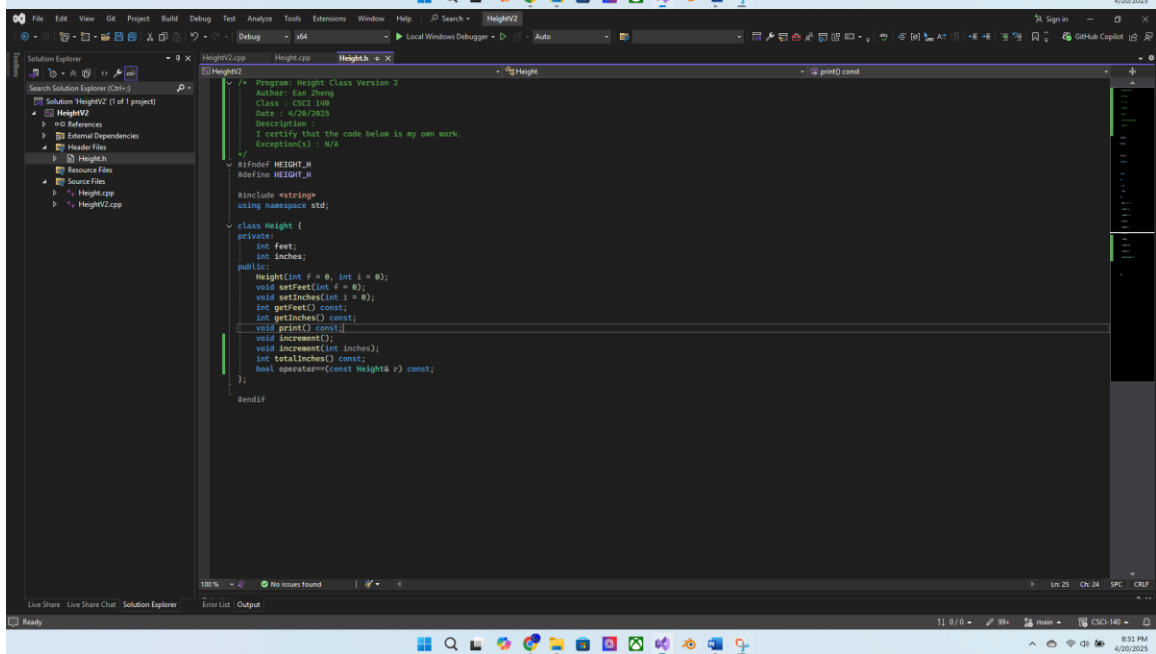
void Height::print() const {
    cout << feet << " " << inches << endl;
}

void Height::increment() {
    ++inches;
    if (inches == 12) {
        inches = 0;
        ++feet;
    }
}

void Height::increment(int inches) {
    if (inches > 0 && inches < 12) {
        this->inches = this->inches + inches;
        if (this->inches == 12) {
            this->inches = 0;
            ++feet;
        }
    }
}

int Height::totalInches() const {
    return (feet * 12) + inches;
}

bool Height::operator==(const Height& r) const {
    if (feet == r.getFeet() && inches == r.getInches())
        return true;
    else if (feet != r.getFeet() || inches != r.getInches())
        return false;
}
```



```
Program: Height Class Version 2
Author: Ean Zhang
Class : CSC2 140
Date : 4/20/2025
Description
I certify that the code below is my own work.
Exception(s) : N/A

#ifndef HEIGHT_H
#define HEIGHT_H

#include <string>
using namespace std;

class Height {
private:
    int feet;
    int inches;
public:
    Height(int f = 0, int i = 0);
    void setFeet(int f = 0);
    void setInches(int i = 0);
    int getFeet() const;
    int getInches() const;
    void print() const;
    void increment();
    void increment(int inches);
    int totalInches() const;
    bool operator==(const Height& r) const;
};

#endif
```

Input/output below:

0' 0"

5' 0"

5' 8"

0' 5"

6' 0"

h3: 5' 8"

feet: 5, inches: 10

feet: 6, inches: 5

h5: 7' 0"

h6: 0' 5"

Total inches : 16

h6: 1' 4"

h3 is not the same as h6

```
Program: Height Class Version 2
Author: Ean Zheng
Class : CSCI 140
Date : 4/26/2025
Description :
I certify that the code below is my own.
Exception(s) : N/A

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

int main() {
    // Create 2 Height objects
    Height h1; // feet: 0, inches: 0
    Height h2(5); // feet: 0, inches: 5
    // Create some Height objects (same as h1)
    Height h3(5, 8); // feet: 5, inches: 8
    Height h4(-3, 15); // feet: 0, inches: 12
    Height h5(6, 15); // feet: 6, inches: 15
    // Print Height h1
    Height array[10]; h1, h2, h3, h4, h5;
    for (int i = 0; i < 5; ++i) {
        array[i].print();
    }
    cout << "h1: ";
    h1.print(); // h1: 0' 0"
    cout << endl;
    // Add more code below to print h4 and
    // Perform various operations
    h3.setFeet(-2); // feet: 5, inches: 8
    h3.setInches(18); // feet: 5, inches:
    cout << "feet: " << h3.getFeet() << ",
    h3.setFeet(0); // feet: 0, inches: 5
    h3.setInches(12); // feet: 0, inches:
    cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() << endl; // 0 5
    h3.setInches(18); // feet: 0, inches: 18
    h3.increment(); // feet: 0, inches: 11
    h3.increment(); // feet: 7, inches: 0
    cout << "h3: ";
    h3.print(); // h3: 7' 0"
    cout << endl;
    //Add more test cases if needed
    Height h6(0, 5); // feet: 0, inches: 5
    // Print height (add code to label height and new line as needed)
    cout << "h6: ";
    h6.print(); // h6: 0' 5"
    h6.increment(12); // ignore, feet: 0, inches: 5
}
```

Question 1: It is possible to use a single int private variable totInches that represents height in inches (instead of variables feet and inches) to implement the Height class without changing the interface (i.e., public member functions stay the same and the application still works the same way). Explain how you would go about doing that.

Yes. I would implement that by adding that every time a new feet or inch value is assigned or upon object initialization, the total Inches value is recalculated given the current/new feet and inches variables. If the increment function is called, the only change would be to add the increment to the variable.

Question 2: List some good reasons for overloading operators in a class.

To reduce work, reduce amount of code, to make code simpler and easier to understand, and make writing the code easier and more convenient.

Extra Credit (2 points): Modify your Height class to add the following operators: + and -. You can add or subtract (absolute difference), and the result would be a Height object. Given two Height objects, h3 and h6, you can do:

```
// assume h3 (5' 8") and h6 (1' 4")
Height h7 = h3 + h6;
Height h8 = h3 - h6;
Height h9 = h6 - h3;
// Print height (add code to label height and new line as needed)
h7.print(); // h7: 7' 0"
// Print height (add code to label height and new line as needed)
h8.print(); // h8: 4' 4"
// Print height (add code to label height and new line as needed)
h9.print(); // h9: 4' 4"
// add more test cases as needed
```

Source code below:

HeightV2.cpp:

```
/* Program: Height Class Version 2 Extra Credit Ver.
   Author: Ean Zheng
   Class : CSCI 140
   Date : 4/20/2025
   Description :
   I certify that the code below is my own work.
   Exception(s) : N/A
*/
#include <iostream>
using namespace std;

#include "Height.h"

int main() {
    // Create 2 Height objects
    Height h1; // feet: 0, inches: 0
    Height h2(5); // feet: 5, inches: 0
    // Create some Height objects (same as original version)
    Height h3(5, 8); // feet: 5, inches: 8
    Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)
    Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)
    // Print height h3
    Height array[] { h1, h2, h3, h4, h5 };
    for (int i = 0; i < 5; ++i) {
        array[i].print();
    }
    cout << "h3: ";
    h3.print(); // h3: 5' 8"
```

```

cout << endl;
// Add more code below to print h4 and h5 like h3 above
// Perform various operations
h3.setFeet(-2); // feet: 5, inches: 8, feet stay the same
h3.setInches(10); // feet: 5, inches: 10
cout << "feet: " << h3.getFeet() << ", inches: " << h3.getInches() << endl; // 5 10
h4.setFeet(6); // feet: 6, inches: 5
h4.setInches(12); // feet: 6, inches: 5, inches stay the same
cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() << endl; // 6 5
h5.setInches(10); // feet: 6, inches: 10
h5.increment(); // feet: 6, inches: 11
h5.increment(); // feet: 7, inches: 0
cout << "h5: ";
h5.print(); // h3: 7' 0"
cout << endl;
//Add more test cases if needed
Height h6(0, 5); // feet: 0, inches: 5
// Print height (add code to label height and new line as needed)
cout << "h6: ";
h6.print(); // h6: 0' 5"
h6.increment(12); // ignore, feet: 0, inches: 5
h6.increment(-3); // ignore, feet: 0, inches: 5
h6.increment(); // feet: 0, inches: 6
h6.increment(10); // feet: 1, inches: 4
cout << "Total inches : " << h6.totalInches() << endl; // 16
// Print height (add code to label height and new line as needed)
cout << "h6: ";
h6.print(); // h6: 1' 4"
if (h3 == h6)
    cout << "h3 is the same as h6" << endl;
else
    cout << "h3 is not the same as h6" << endl;
// should output: h3 is not the same as h6
// assume h3 (5' 8") and h6 (1' 4")
h3.setInches(8);
Height h7 = h3 + h6;
Height h8 = h3 - h6;
Height h9 = h6 - h3;
// Print height (add code to label height and new line as needed)
cout << "h7: ";
h7.print(); // h7: 7' 0"
// Print height (add code to label height and new line as needed)
cout << "h8: ";
h8.print(); // h8: 4' 4"
// Print height (add code to label height and new line as needed)
cout << "h9: ";

```

```
    h9.print(); // h9: 4' 4"  
    // add more test cases as needed  
}
```

Height.cpp:

```
/* Program: Height Class Version 2 Extra Credit Ver.
```

```
   Author: Ean Zheng
```

```
   Class : CSCI 140
```

```
   Date : 4/20/2025
```

```
   Description :
```

```
   I certify that the code below is my own work.
```

```
   Exception(s) : N/A
```

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
#include "Height.h"
```

```
Height::Height(int f, int i) : feet(f), inches(i) {
```

```
    if (feet < 0) feet = 0;
```

```
    if (inches < 0 || inches > 11) inches = 0;
```

```
}
```

```
void Height::setFeet(int f) {
```

```
    if (f >= 0) feet = f;
```

```
}
```

```
void Height::setInches(int i) {
```

```
    if (i >= 0 && i <= 11) inches = i;
```

```
}
```

```
int Height::getFeet() const {
```

```
    return feet;
```

```
}
```

```
int Height::getInches() const {
```

```
    return inches;
```

```
}
```

```
void Height::print() const {
```

```
    cout << feet << " " << inches << "\"\" << endl;
```

```
}
```

```
void Height::increment() {
```

```
    ++inches;
```

```
    if (inches == 12) {
```

```
        inches = 0;
```

```

        ++feet;
    }
}

void Height::increment(int inches) {
    if (inches > 0 && inches < 12) {
        this->inches = this->inches + inches;
        if (this->inches >= 12) {
            this->inches -= 12;
            ++feet;
        }
    }
}

int Height::totalInches() const {
    return (feet * 12) + inches;
}

bool Height::operator==(const Height& r) const {
    if (feet == r.getFeet() && inches == r.getInches())
        return true;
    else if (feet != r.getFeet() || inches != r.getInches())
        return false;
}

Height Height::operator+(Height rhs) {
    int tempfeet = feet + rhs.getFeet();
    int tempinches = inches + rhs.getInches();
    if (tempinches >= 12) {
        tempinches -= 12;
        ++tempfeet;
    }
    Height result(tempfeet, tempinches);
    return result;
}

Height Height::operator-(Height rhs) {
    int tempfeet = max(feet, rhs.getFeet()) - min(feet, rhs.getFeet());
    int tempinches = max(inches, rhs.getInches()) - min(inches, rhs.getInches());
    if (tempinches < 0) {
        tempinches = 12 + tempinches;
        --tempfeet;
    }
    Height result(tempfeet, tempinches);
    return result;
}

```

Height.h:

```
/* Program: Height Class Version 2 Extra Credit Ver.
```

```
   Author: Ean Zheng
```

```
   Class : CSCI 140
```

```
   Date : 4/20/2025
```

```
   Description :
```

```
   I certify that the code below is my own work.
```

```
   Exception(s) : N/A
```

```
*/
```

```
#ifndef HEIGHT_H
```

```
#define HEIGHT_H
```

```
#include <string>
```

```
using namespace std;
```

```
class Height {
```

```
private:
```

```
    int feet;
```

```
    int inches;
```

```
public:
```

```
    Height(int f = 0, int i = 0);
```

```
    void setFeet(int f = 0);
```

```
    void setInches(int i = 0);
```

```
    int getFeet() const;
```

```
    int getInches() const;
```

```
    void print() const;
```

```
    void increment();
```

```
    void increment(int inches);
```

```
    int totalInches() const;
```

```
    bool operator==(const Height& r) const;
```

```
    Height operator+(Height rhs);
```

```
    Height operator-(Height rhs);
```

```
};
```

```
#endif
```


Input/output below:

0' 0"

5' 0"

5' 8"

0' 5"

6' 0"

h3: 5' 8"

feet: 5, inches: 10

feet: 6, inches: 5

h5: 7' 0"

h6: 0' 5"

Total inches : 16

h6: 1' 4"

h3 is not the same as h6

h7: 7' 0"

h8: 4' 4"

h9: 4' 4"

```
// Program: Height Class Version 2 Extra Credit Ver.
// Author: Kan Zheng
// Class : CSCI 140
// Date : 4/26/2025
// Description:
// I certify that the code below is my own work.
// Exception(s) : N/A

#include <iostream>
using namespace std;

#include "Height.h"

int main() {
    // Create 2 Height objects
    Height h1; // feet: 0, inches: 0
    Height h2(5); // feet: 5, inches: 0
    // Create same Height objects (same as original ver)
    Height h3(5, 8); // feet: 5, inches: 8
    Height h4(-5, 5); // feet: 0, inches: 5 (invalid for
    Height h5(6, 15); // feet: 6, inches: 0 (invalid for
    // Print Height h3
    Height array[10] { h1, h2, h3, h4, h5 };
    for (int i = 0; i < 5; ++i) {
        array[i].print();
    }
    cout << endl;
    // Add more code below to print h4 and h5 line h3 as
    // Perform various operations
    h3.setFeet(-2); // feet: 0, inches: 8, feet stay the
    h3.setInches(10); // feet: 0, inches: 10
    cout << "feet: " << h3.getFeet() << ", inches: " <<
    h4.setFeet(0); // feet: 0, inches: 5
    h4.setInches(12); // feet: 0, inches: 5, inches stay
    cout << "feet: " << h4.getFeet() << ", inches: " << h4.getInches() << endl; // 0 5
    h5.setInches(10); // feet: 6, inches: 10
    h5.increment(); // feet: 6, inches: 11
    h5.increment(); // feet: 7, inches: 0
    cout << endl;
    h5.print(); // h5: 7' 0"
    cout << endl;
    // Add more test cases if needed
    Height h6(0, 5); // feet: 0, inches: 5
    // Print height (add code to label height and new line as needed)
    cout << "h6: ";
    h6.print(); // h6: 0' 5"
    h6.increment(12); // ignore, feet: 0, inches: 5
    cout << endl;
}
```

0' 0"
5' 0"
5' 8"
0' 5"
6' 0"
h3: 5' 8"
feet: 5, inches: 10
feet: 6, inches: 5
h5: 7' 0"
h6: 0' 5"
Total inches : 16
h6: 1' 4"
h3 is not the same as h6
h7: 7' 0"
h8: 4' 4"
h9: 4' 4"

C:\GithubRepos\CSCI-140\Programs\V5ShoppingCart\HeightV2\Debug\HeightV2.exe (process 36188) exited with code 0 (0x0)
Press any key to close this window . . .