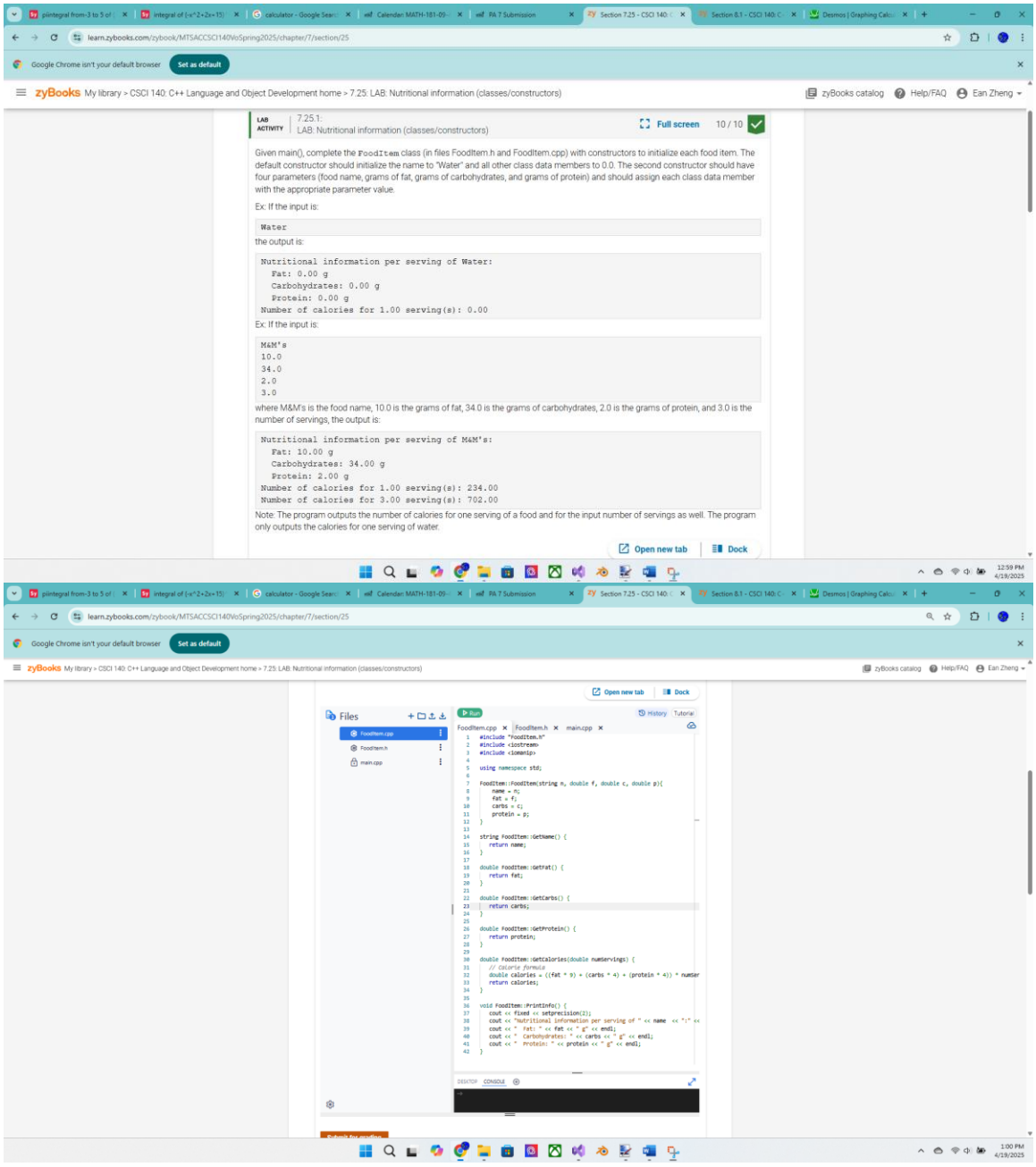
# CSCI 140 PA 7 Submission

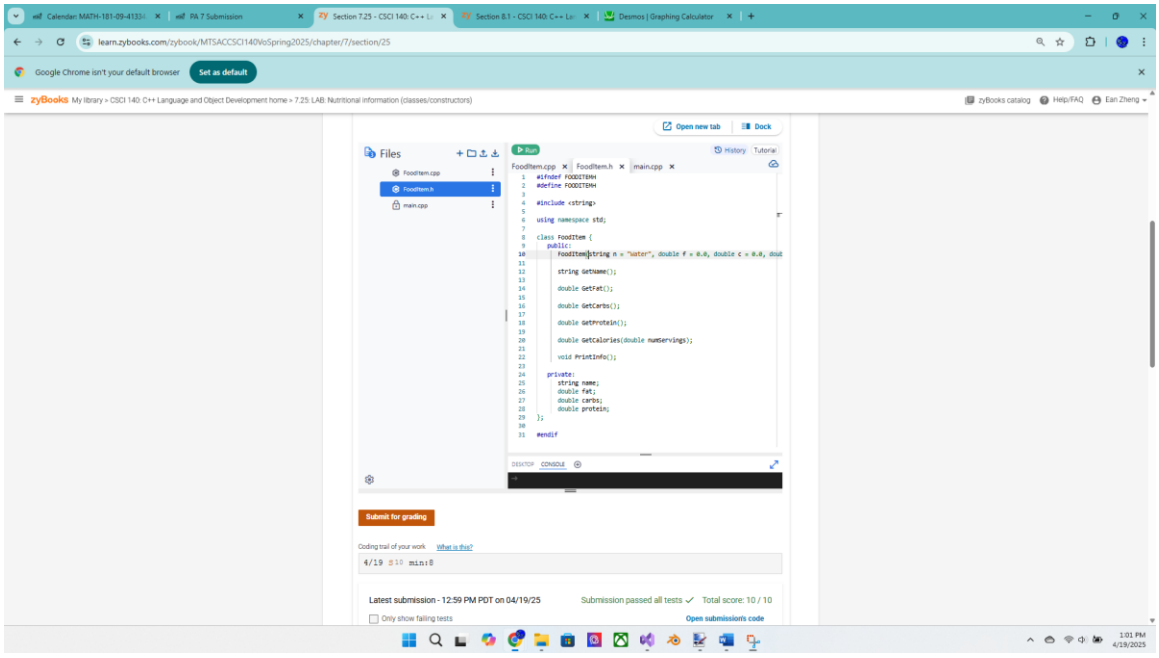
Due Date: 4/21/2025

Name(s): Ean Zheng

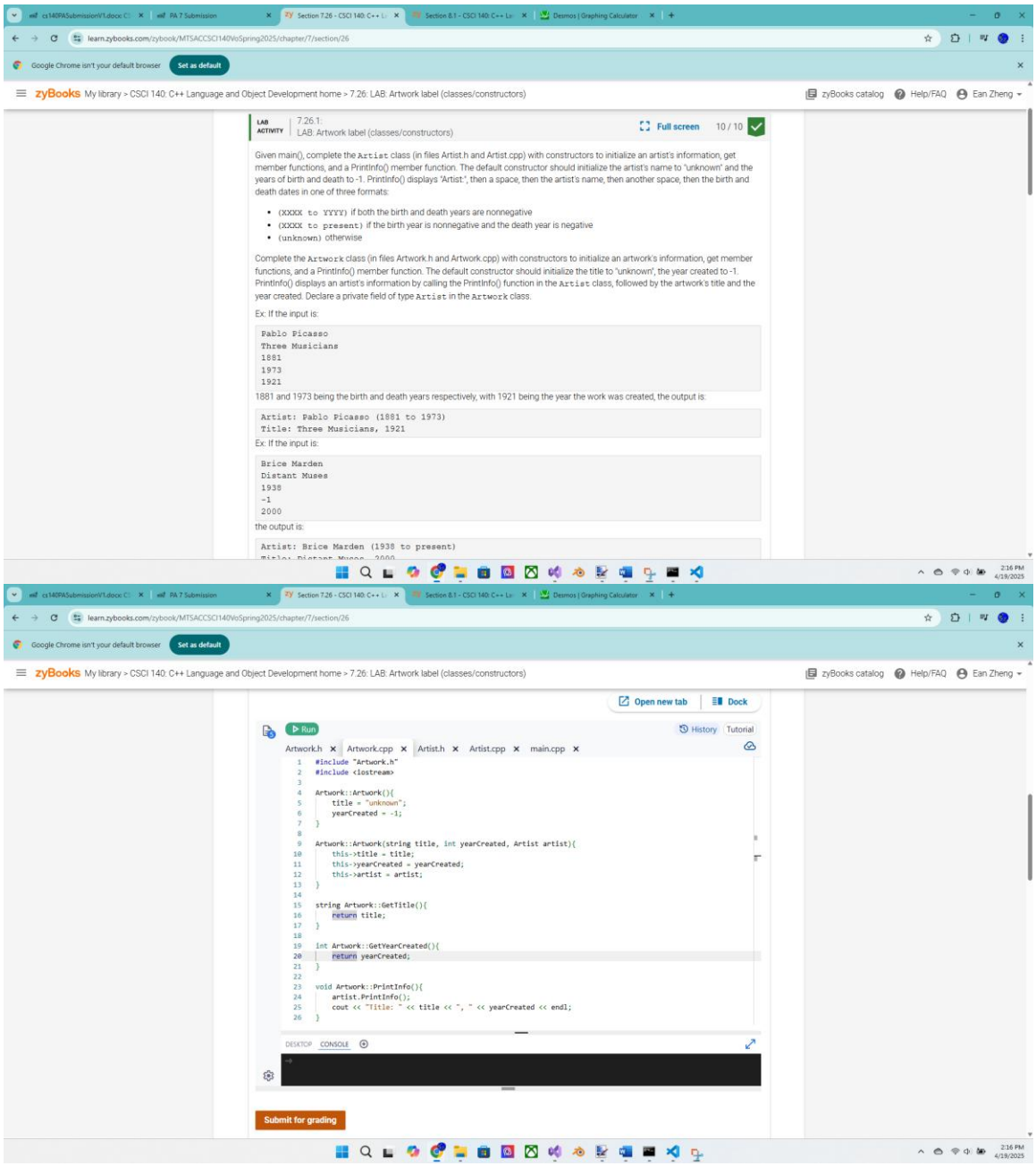
---

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




Open new tab
Dock
Files
FoodItem.cpp
FoodItem.h
main.cpp
Run
History
Tutorial
FoodItem.cpp
FoodItem.h
main.cpp
#ifndef FOODITEMH
#define FOODITEMH
#include <string>
using namespace std;
class FoodItem {
public:
FoodItem(string n = "water", double f = 0.0, double c = 0.0, doub
string GetName();
double GetFat();
double GetCarbs();
double GetProtein();
double GetCalories(double numServings);
void PrintInfo();
private:
string name;
double fat;
double carbs;
double protein;
};
#endif
DESKTOP
CONSOLE
Submit for grading
Coding trail of your work
What is this?
4/19 10 min:8
Latest submission - 12:59 PM PDT on 04/19/25
Submission passed all tests
Total score: 10 / 10
Only show failing tests
Open submission's code

# Exercise 2 – 7.26 LAB: Artwork label



LAB ACTIVITY | 7.26.1: LAB: Artwork label (classes/constructors) — Full screen — 10 / 10

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)
```



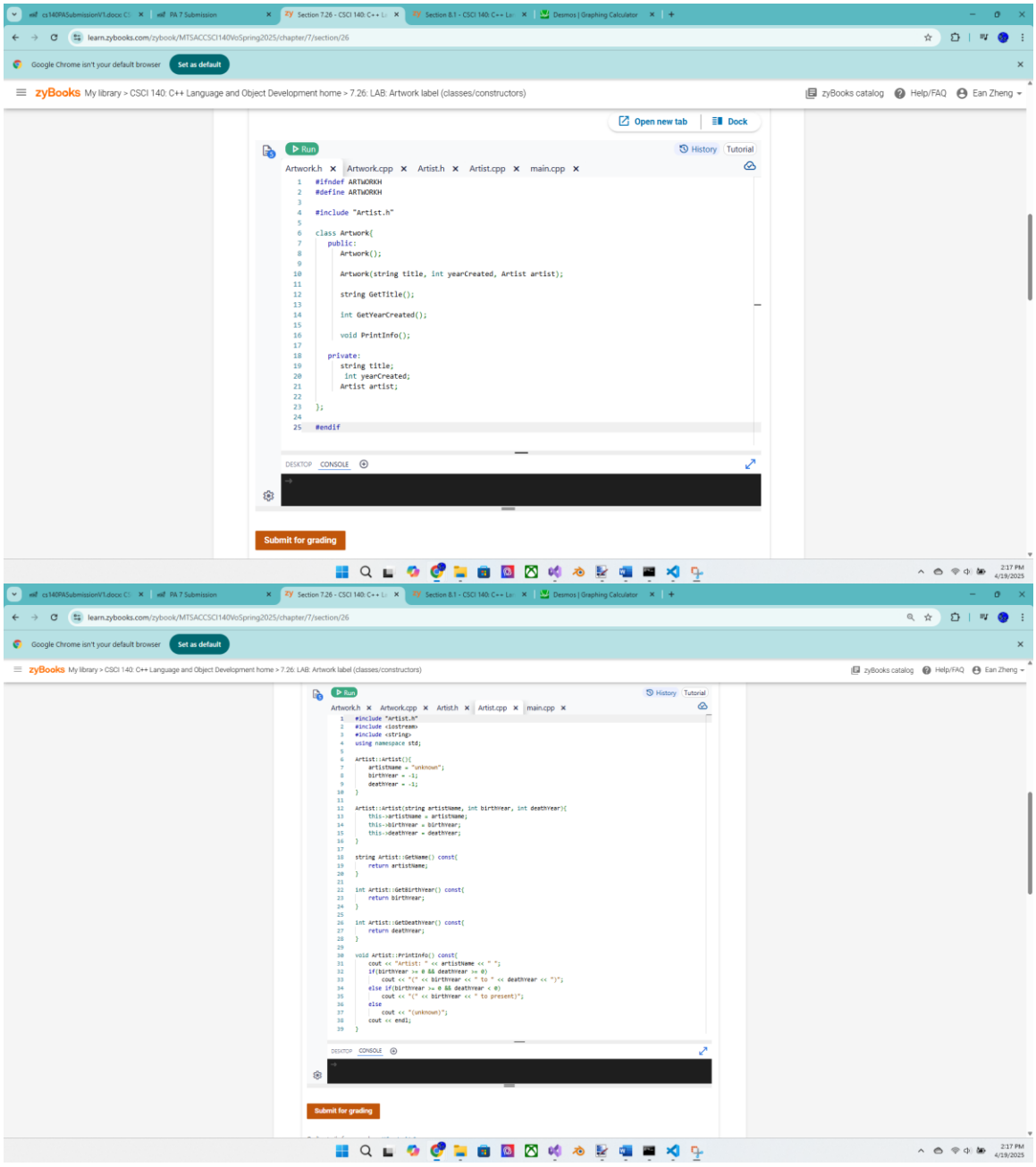
Open new tab | Dock — Run — History | Tutorial

Artwork.h | Artwork.cpp | Artist.h | Artist.cpp | main.cpp

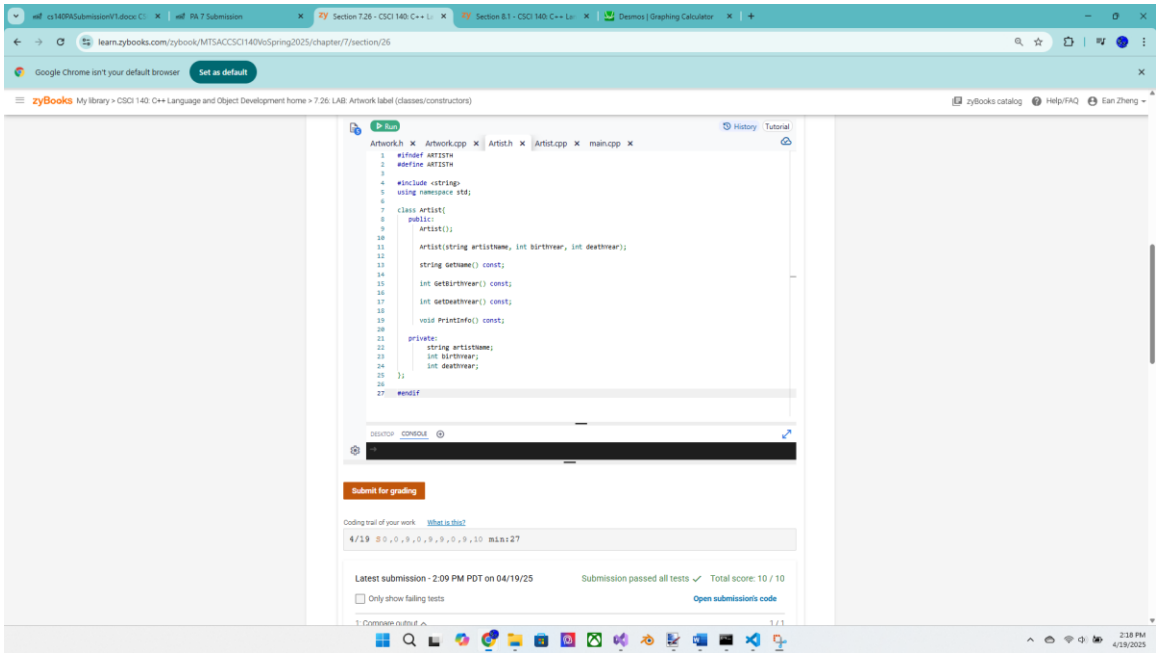
```
#include "Artwork.h"
#include <iostream>

Artwork::Artwork(){
    title = "unknown";
    yearCreated = -1;
}

Artwork::Artwork(string title, int yearCreated, Artist artist){
    this->title = title;
    this->yearCreated = yearCreated;
    this->artist = artist;
}

string Artwork::GetTitle(){
    return title;
}

int Artwork::GetYearCreated(){
    return yearCreated;
}

void Artwork::PrintInfo(){
    artist.PrintInfo();
    cout << "Title: " << title << ", " << yearCreated << endl;
}
```

DESKTOP | CONSOLE

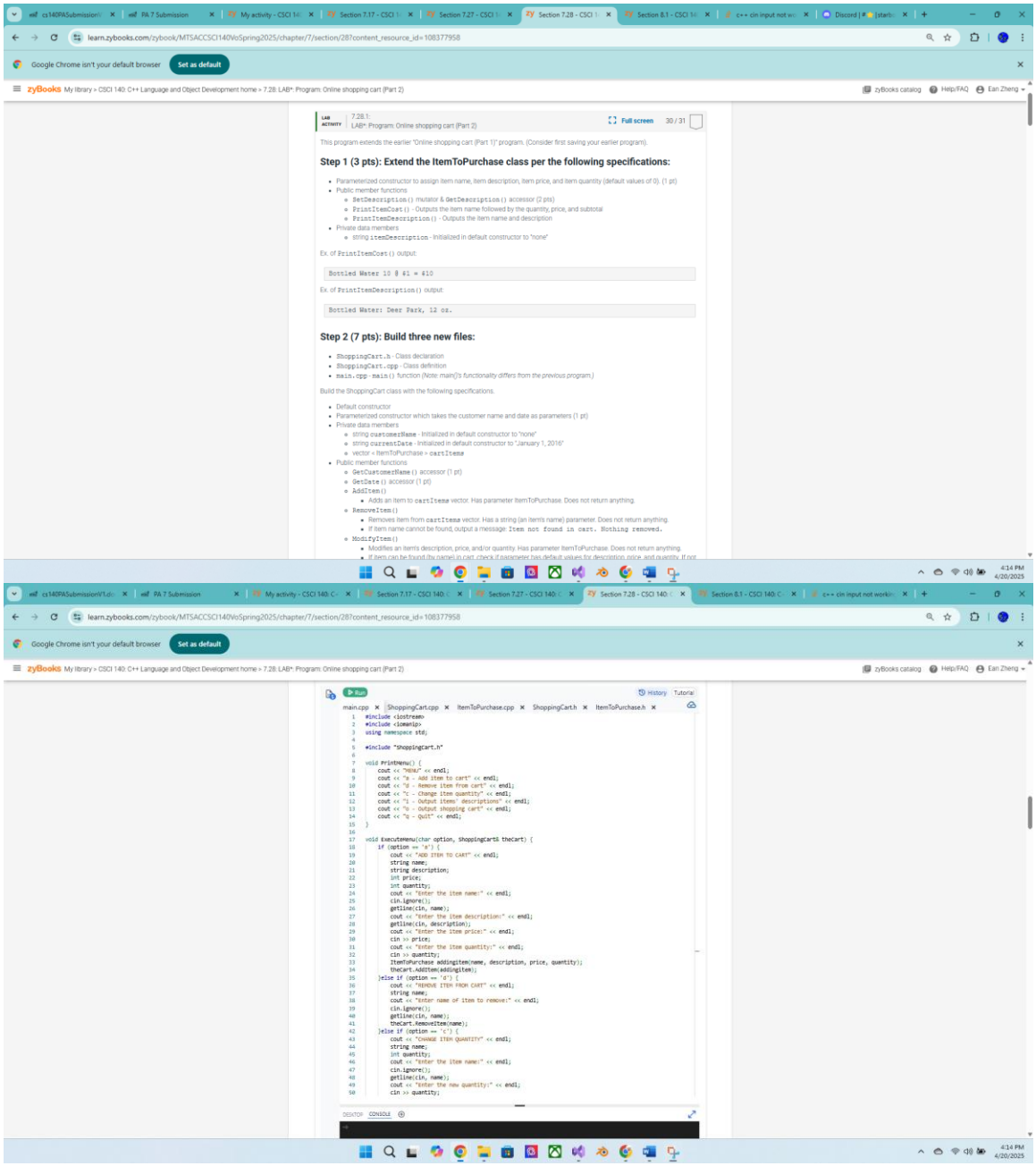
Submit for grading


learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/chapter/7/section/26
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
History
Tutorial
Artwork.h
Artwork.cpp
Artist.h
Artist.cpp
main.cpp
#ifndef ARTWORKH
#define ARTWORKH
#include "Artist.h"
class Artwork{
public:
Artwork();
Artwork(string title, int yearCreated, Artist artist);
string GetTitle();
int GetYearCreated();
void PrintInfo();
private:
string title;
int yearCreated;
Artist artist;
};
#endif
DESKTOP
CONSOLE
Submit for grading
2:17 PM
4/19/2025


learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/chapter/7/section/26
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
History
Tutorial
Artwork.h
Artwork.cpp
Artist.h
Artist.cpp
main.cpp
#include "Artist.h"
#include <iostream>
#include <string>
using namespace std;
Artist::Artist(){
artistName = "unknown";
birthYear = -1;
deathYear = -1;
}
Artist::Artist(string artistName, int birthYear, int deathYear){
this->artistName = artistName;
this->birthYear = birthYear;
this->deathYear = deathYear;
}
string Artist::GetName() const{
return artistName;
}
int Artist::GetBirthYear() const{
return birthYear;
}
int Artist::GetDeathYear() const{
return deathYear;
}
void Artist::PrintInfo() const{
cout << "Artist: " << artistName << " ";
if(birthYear >= 0 && deathYear >= 0)
cout << "(" << birthYear << " to " << deathYear << ")";
else if(birthYear >= 0 && deathYear < 0)
cout << "(" << birthYear << " to present)";
else
cout << "(unknown)";
cout << endl;
}
DESKTOP
CONSOLE
Submit for grading
2:17 PM
4/19/2025


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
History
Tutorial
Artwork.h
Artwork.cpp
Artist.h
Artist.cpp
main.cpp
#ifndef ARTISTH
#define ARTISTH
#include <string>
using namespace std;
class Artist{
public:
Artist();
Artist(string artistName, int birthYear, int deathYear);
string GetName() const;
int GetBirthYear() const;
int GetDeathYear() const;
void PrintInfo() const;
private:
string artistName;
int birthYear;
int deathYear;
};
#endif
DESKTOP
CONSOLE
Submit for grading
Coding trail of your work
What is this?
4/19 80,0,9,0,9,9,0,9,10 min: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: Compare 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



LAB ACTIVITY 7.28.1: LAB*: Program: Online shopping cart (Part 2)

This program extends the earlier "Online shopping cart (Part 1)" program. (Consider first saving your earlier program).

**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()'s functionality differs from the previous program.)*

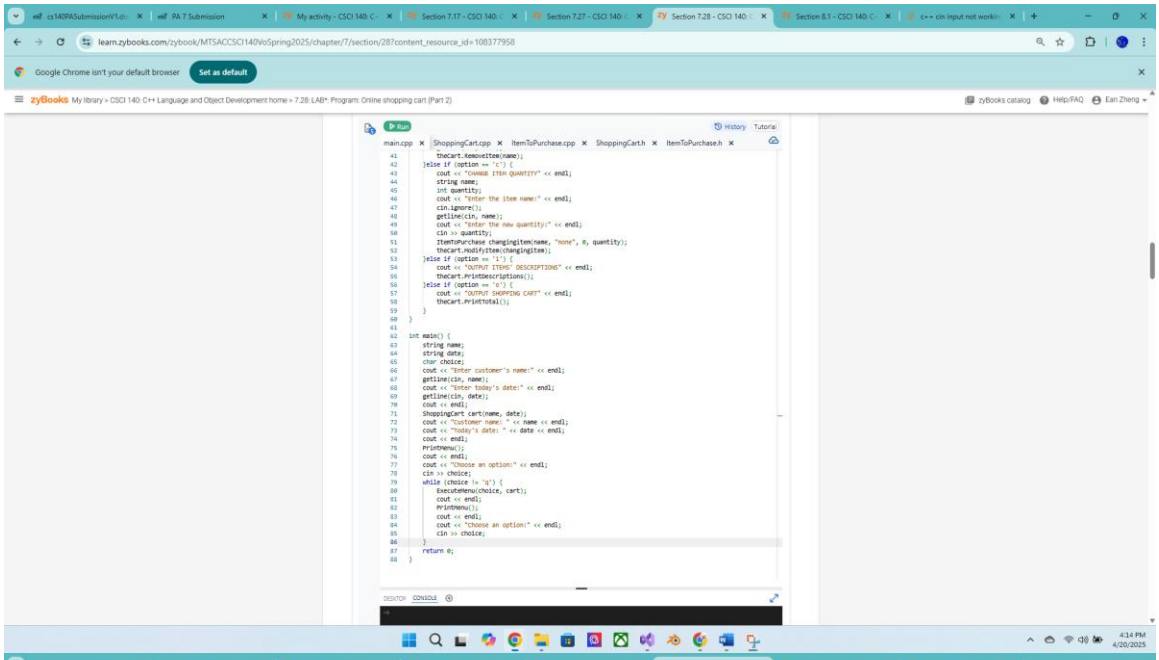
Build the ShoppingCart class with the following specifications.

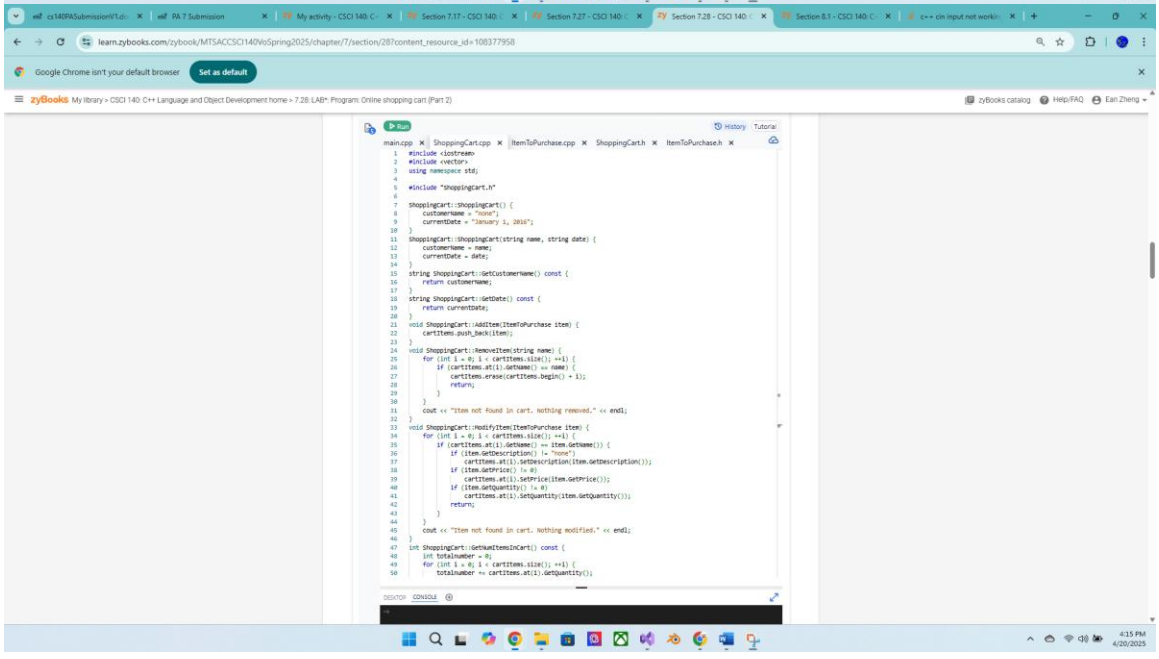
- Default constructor
- Parameterized constructor which takes the customer name and date as parameters (1 pt)
- Private data members
  - string customerName - Initialized in default constructor to "none"
  - string currentDate - Initialized in default constructor to "January 1, 2016"
  - vector < ItemToPurchase > 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 (by name) in cart, check if parameter has default values for description, price, and quantity. If not

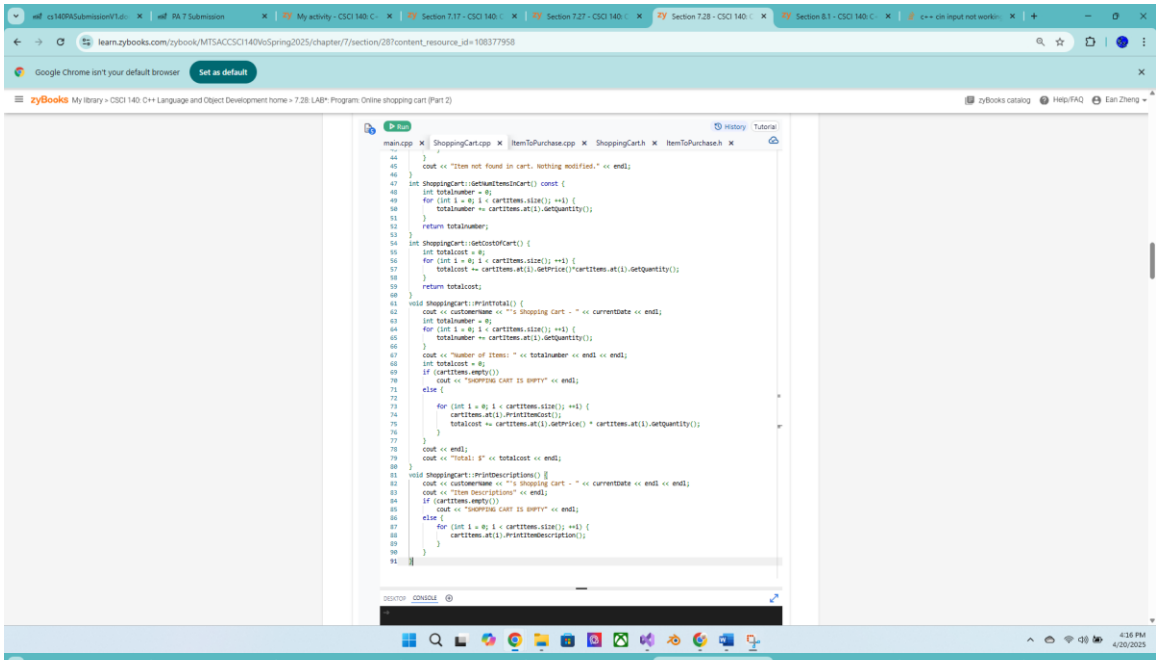


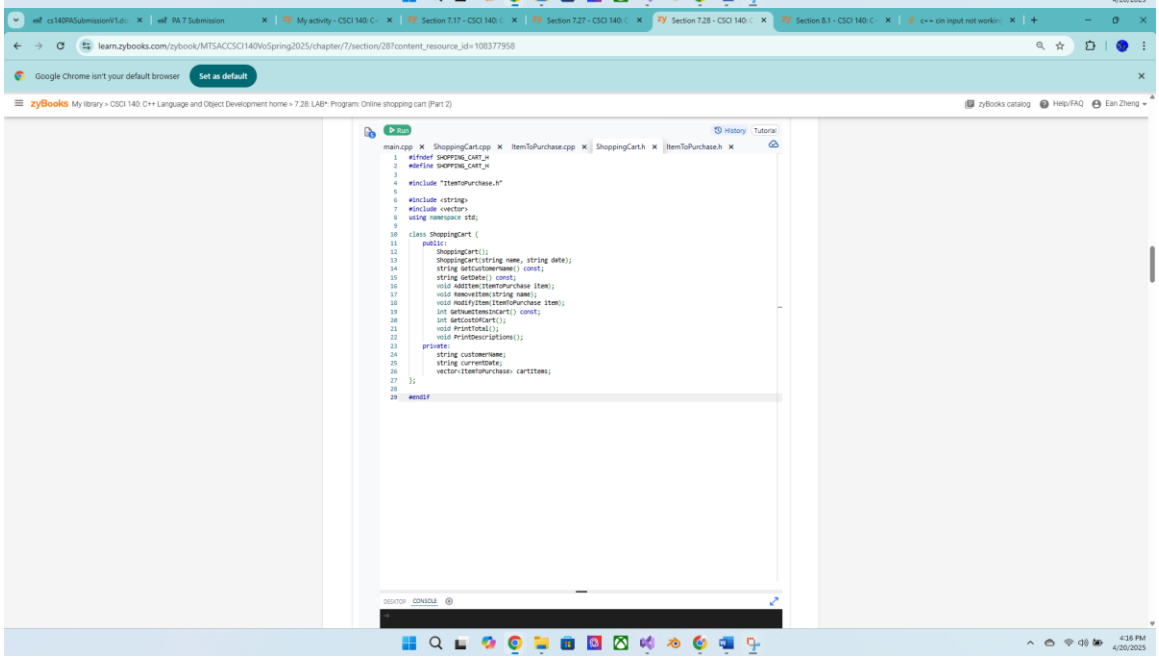
main.cpp | ShoppingCart.cpp | ItemToPurchase.cpp | ShoppingCart.h | ItemToPurchase.h

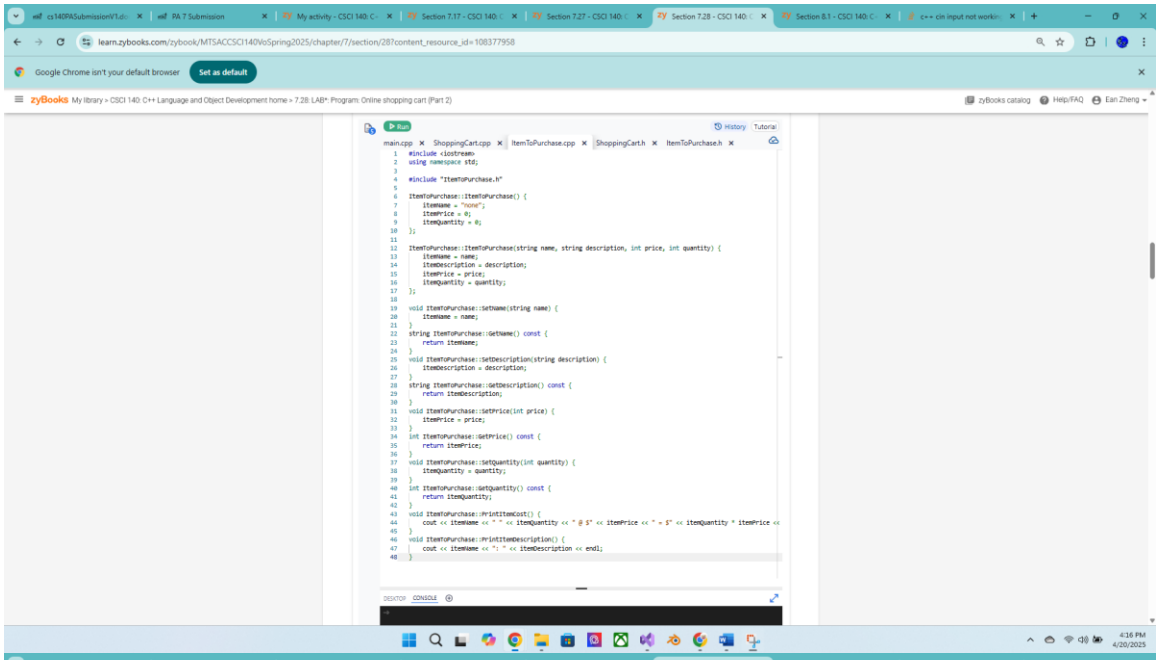
```
#include <iostream>
#include <iomanip>
using namespace std;

#include "ShoppingCart.h"

void PrintMenu() {
   cout << "MENU" << endl;
   cout << "a - Add item to cart" << endl;
   cout << "d - Remove item from cart" << endl;
   cout << "c - Change item quantity" << endl;
   cout << "i - Output items' descriptions" << endl;
   cout << "o - Output shopping cart" << endl;
   cout << "q - Quit" << endl;
}

void ExecuteMenu(char option, ShoppingCart& theCart) {
   if (option == 'a') {
      cout << "ADD ITEM TO CART" << endl;
      string name;
      string description;
      int price;
      int quantity;
      cout << "Enter the item name:" << endl;
      cin.ignore(1);
      getline(cin, name);
      cout << "Enter the item description:" << endl;
      getline(cin, description);
      cout << "Enter the item price:" << endl;
      cin >> price;
      cout << "Enter the item quantity:" << endl;
      cin >> quantity;
      ItemToPurchase additem(name, description, price, quantity);
      theCart.AddItem(additem);
   }else if (option == 'd') {
      cout << "REMOVE ITEM FROM CART" << endl;
      string name;
      cout << "Enter name of item to remove:" << endl;
      cin.ignore(1);
      getline(cin, name);
      theCart.RemoveItem(name);
   }else if (option == 'c') {
      cout << "CHANGE ITEM QUANTITY" << endl;
      string name;
      int quantity;
      cout << "Enter the item name:" << endl;
      cin.ignore(1);
      getline(cin, name);
      cout << "Enter the new quantity:" << endl;
      cin >> quantity;
```

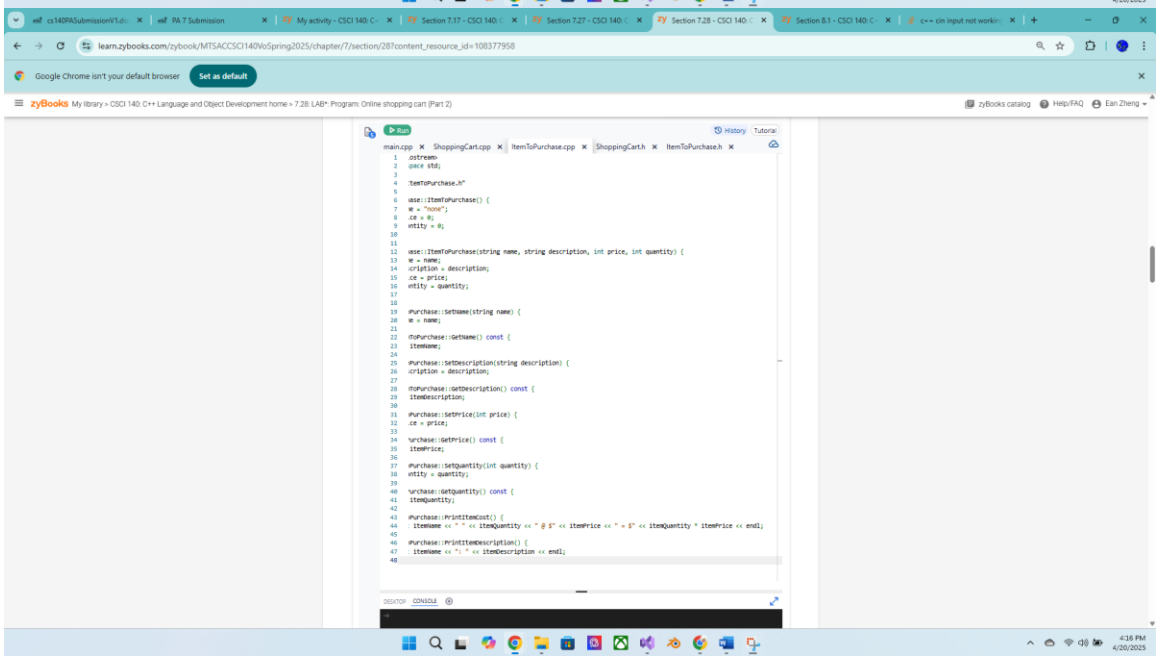

learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/chapter/7/section/28?content_resource_id=108377958
Google Chrome isn't your default browser
Set as default
zyBooks My library > CSCI 140: C++ Language and Object Development home > 7.28: LAB*: Program: Online shopping cart (Part 2)
zyBooks catalog Help/FAQ Ean Zheng
Run
History Tutorial
main.cpp ShoppingCart.cpp ItemToPurchase.cpp ShoppingCart.h ItemToPurchase.h
theCart.RemoveItem(name);
}else if (option == 'c') {
cout << "CHANGE ITEM QUANTITY" << endl;
string name;
int quantity;
cout << "Enter the item name:" << endl;
cin.ignore(1);
getline(cin, name);
cout << "Enter the new quantity:" << endl;
cin >> quantity;
ItemToPurchase changingItem(name, "none", 0, quantity);
theCart.ModifyItem(changingItem);
}else if (option == 'i') {
cout << "OUTPUT ITEMS' DESCRIPTIONS" << endl;
theCart.PrintDescriptions();
}else if (option == 'o') {
cout << "OUTPUT SHOPPING CART" << endl;
theCart.PrintTotal();
}
}
int main() {
string name;
string date;
char choice;
cout << "Enter customer's name:" << endl;
getline(cin, name);
cout << "Enter today's date:" << endl;
getline(cin, date);
cout << endl;
ShoppingCart cart(name, date);
cout << "Customer name: " << name << endl;
cout << "Today's date: " << date << endl;
cout << endl;
PrintMenu();
cout << endl;
cout << "Choose an option:" << endl;
cin >> choice;
while (choice != 'q') {
ExecuteMenu(choice, cart);
cout << endl;
PrintMenu();
cout << endl;
cout << "Choose an option:" << endl;
cin >> choice;
}
return 0;
}
DESKTOP CONSOLE
4:14 PM
4/20/2025

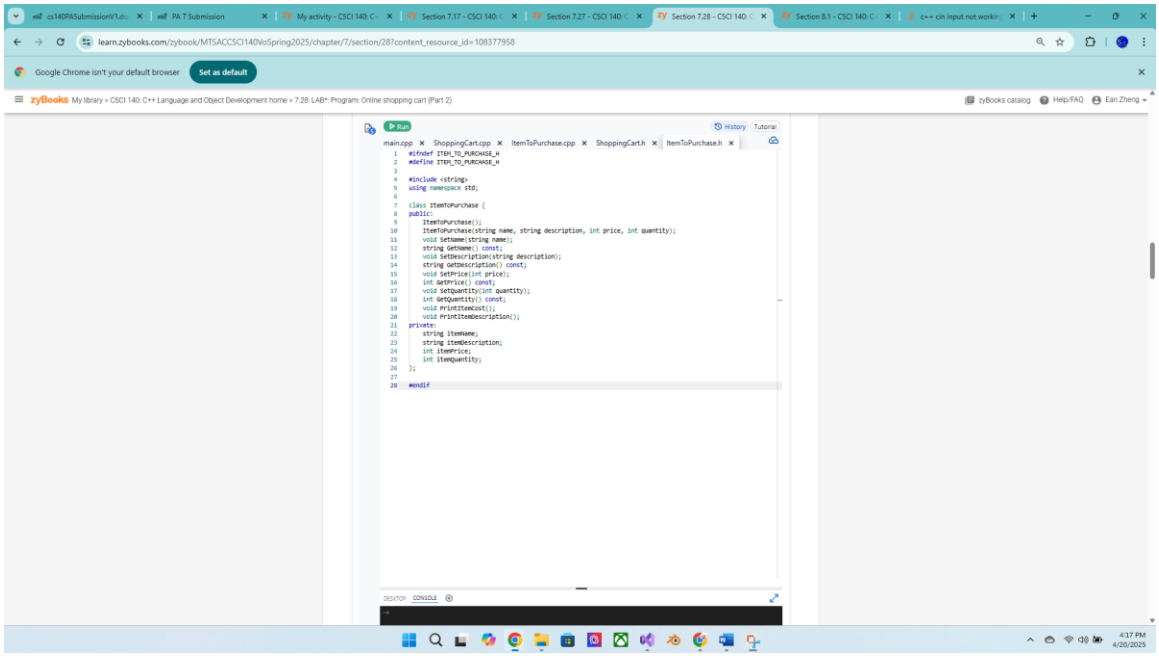

learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/chapter/7/section/28?content_resource_id=108377958
Google Chrome isn't your default browser
Set as default
zyBooks My library > CSCI 140: C++ Language and Object Development home > 7.28: LAB*: Program: Online shopping cart (Part 2)
zyBooks catalog Help/FAQ Ean Zheng
Run
History Tutorial
main.cpp ShoppingCart.cpp ItemToPurchase.cpp ShoppingCart.h ItemToPurchase.h
#include <iostream>
#include <vector>
using namespace std;
#include "ShoppingCart.h"
ShoppingCart::ShoppingCart() {
customerName = "none";
currentDate = "January 1, 2016";
}
ShoppingCart::ShoppingCart(string name, string date) {
customerName = name;
currentDate = date;
}
string ShoppingCart::GetCustomerName() const {
return customerName;
}
string ShoppingCart::GetDate() const {
return currentDate;
}
void ShoppingCart::AddItem(ItemToPurchase item) {
cartItems.push_back(item);
}
void ShoppingCart::RemoveItem(string name) {
for (int i = 0; i < cartItems.size(); ++i) {
if (cartItems.at(i).GetName() == name) {
cartItems.erase(cartItems.begin() + i);
return;
}
}
cout << "Item not found in cart. Nothing removed." << endl;
}
void ShoppingCart::ModifyItem(ItemToPurchase item) {
for (int i = 0; i < cartItems.size(); ++i) {
if (cartItems.at(i).GetName() == item.GetName()) {
if (item.GetDescription() != "none")
cartItems.at(i).SetDescription(item.GetDescription());
if (item.GetPrice() != 0)
cartItems.at(i).SetPrice(item.GetPrice());
if (item.GetQuantity() != 0)
cartItems.at(i).SetQuantity(item.GetQuantity());
return;
}
}
cout << "Item not found in cart. Nothing modified." << endl;
}
int ShoppingCart::GetNumItemsInCart() const {
int totalNumber = 0;
for (int i = 0; i < cartItems.size(); ++i) {
totalNumber += cartItems.at(i).GetQuantity();
DESKTOP CONSOLE
4:15 PM
4/20/2025


learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/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
Run
History
Tutorial
main.cpp
ShoppingCart.cpp
ItemToPurchase.cpp
ShoppingCart.h
ItemToPurchase.h
DESKTOP
CONSOLE
4:16 PM
4/20/2025


learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/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
Run
History
Tutorial
main.cpp
ShoppingCart.cpp
ItemToPurchase.cpp
ShoppingCart.h
ItemToPurchase.h
DESKTOP
CONSOLE
4:16 PM
4/20/2025

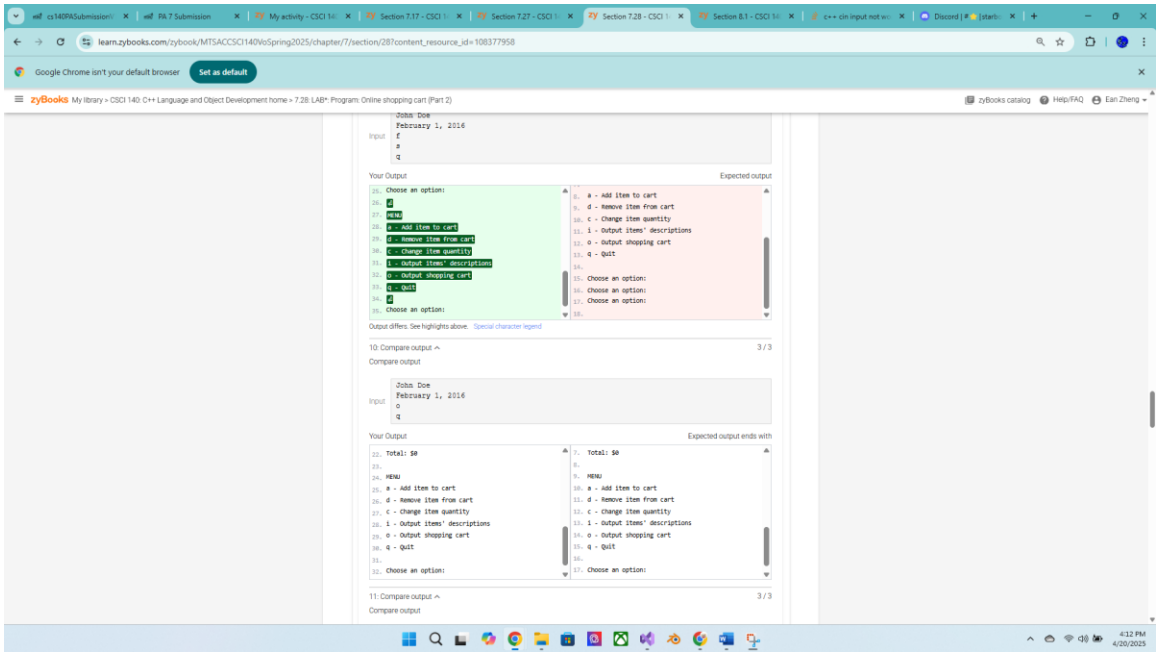

learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/chapter/7/section/28?content_resource_id=108377958
Google Chrome isn't your default browser
Set as default
zyBooks My library > CSCI 140: C++ Language and Object Development home > 7.28: LAB*: Program: Online shopping cart (Part 2)
zyBooks catalog
Help/FAQ
Ean Zheng
Run
History
Tutorial
main.cpp
ShoppingCart.cpp
ItemToPurchase.cpp
ShoppingCart.h
ItemToPurchase.h
#include <iostream>
using namespace std;
#include "ItemToPurchase.h"
ItemToPurchase::ItemToPurchase() {
itemName = "none";
itemPrice = 0;
itemQuantity = 0;
};
ItemToPurchase::ItemToPurchase(string name, string description, int price, int quantity) {
itemName = name;
itemDescription = description;
itemPrice = price;
itemQuantity = quantity;
};
void ItemToPurchase::SetName(string name) {
itemName = name;
}
string ItemToPurchase::GetName() const {
return itemName;
}
void ItemToPurchase::SetDescription(string description) {
itemDescription = description;
}
string ItemToPurchase::GetDescription() const {
return itemDescription;
}
void ItemToPurchase::SetPrice(int price) {
itemPrice = price;
}
int ItemToPurchase::GetPrice() const {
return itemPrice;
}
void ItemToPurchase::SetQuantity(int quantity) {
itemQuantity = quantity;
}
int ItemToPurchase::GetQuantity() const {
return itemQuantity;
}
void ItemToPurchase::PrintItemCost() {
cout << itemName << " " << itemQuantity << " @ $" << itemPrice << " = $" << itemQuantity * itemPrice << endl;
}
void ItemToPurchase::PrintItemDescription() {
cout << itemName << ": " << itemDescription << endl;
}
DESKTOP
CONSOLE
4:16 PM
4/20/2025


learn.zybooks.com/zybook/MTSACCSCI140VoSpring2025/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
Run
History
Tutorial
main.cpp
ShoppingCart.cpp
ItemToPurchase.cpp
ShoppingCart.h
ItemToPurchase.h
DESKTOP
CONSOLE
4:16 PM
4/20/2025

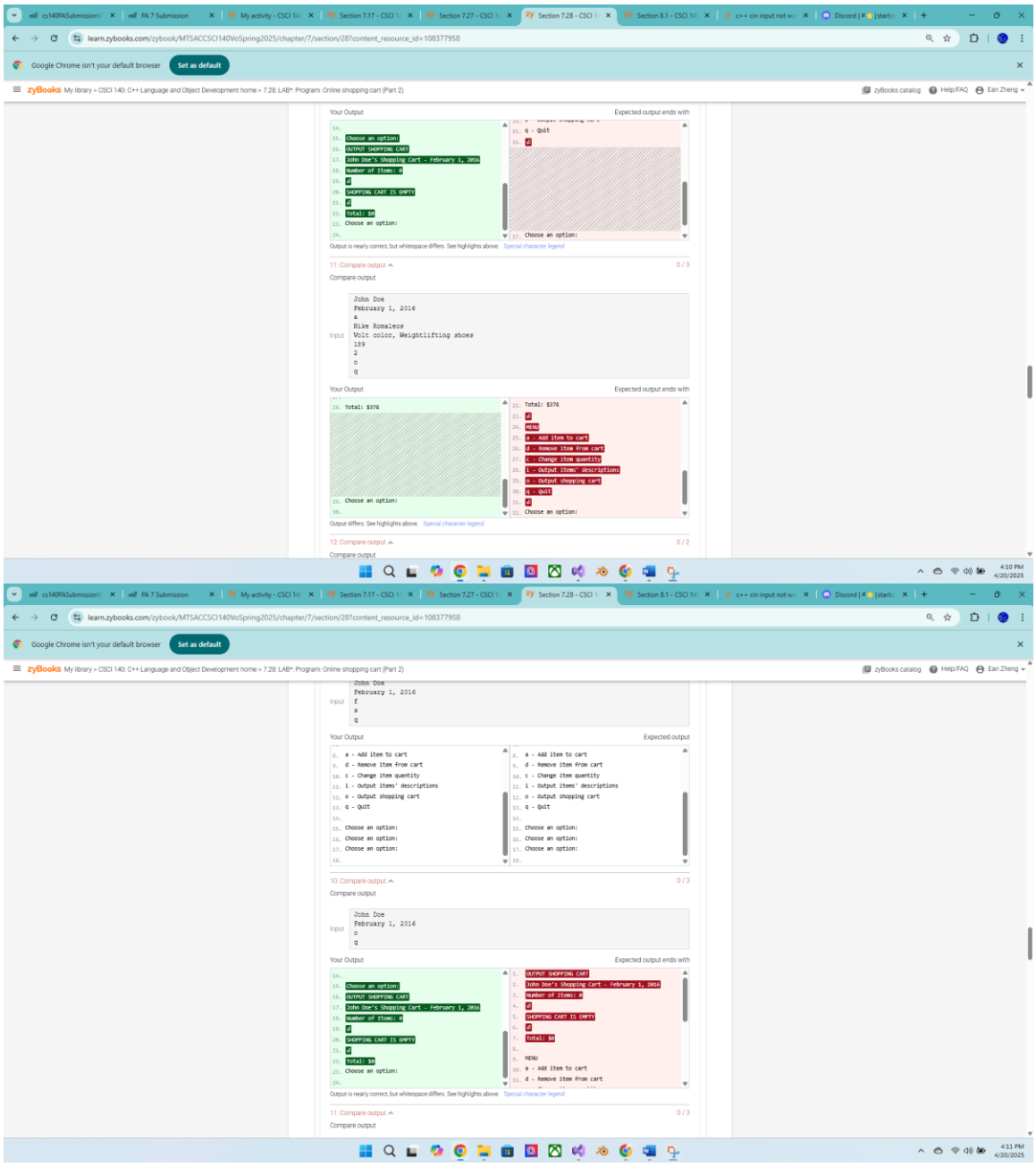


1 input case can’t be satisfied with the rest at the same time, the rest require printing menu every choice loop iteration while this one doesn’t; is website issue

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:



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(); // 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
}
```

Height.cpp:

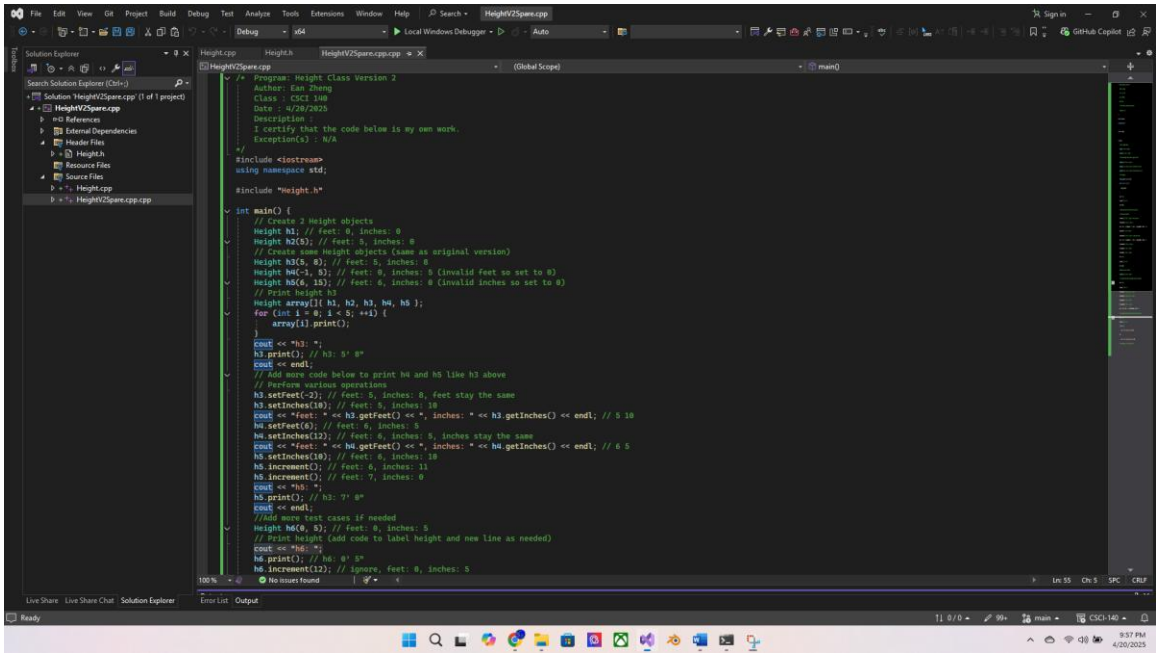
```
/*  Program: Height Class Version 2
    Author: Ean Zheng
    Class : CSCI 140
    Date : 4/20/2025
    Description :
    I certify that the code below is my own work.
    Exception(s) : N/A
*/
#include <iostream>
using namespace std;

#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;
}
```

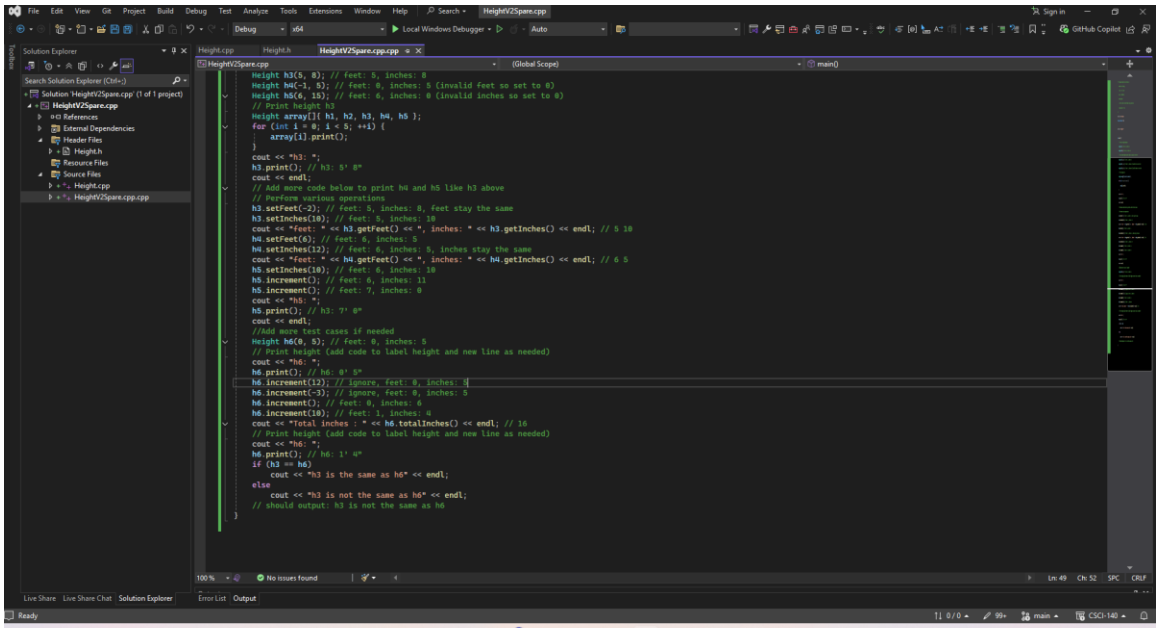
```
void Height::increment() {
   ++inches;
   if (inches == 12) {
      inches = 0;
      ++feet;
   }
}

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:

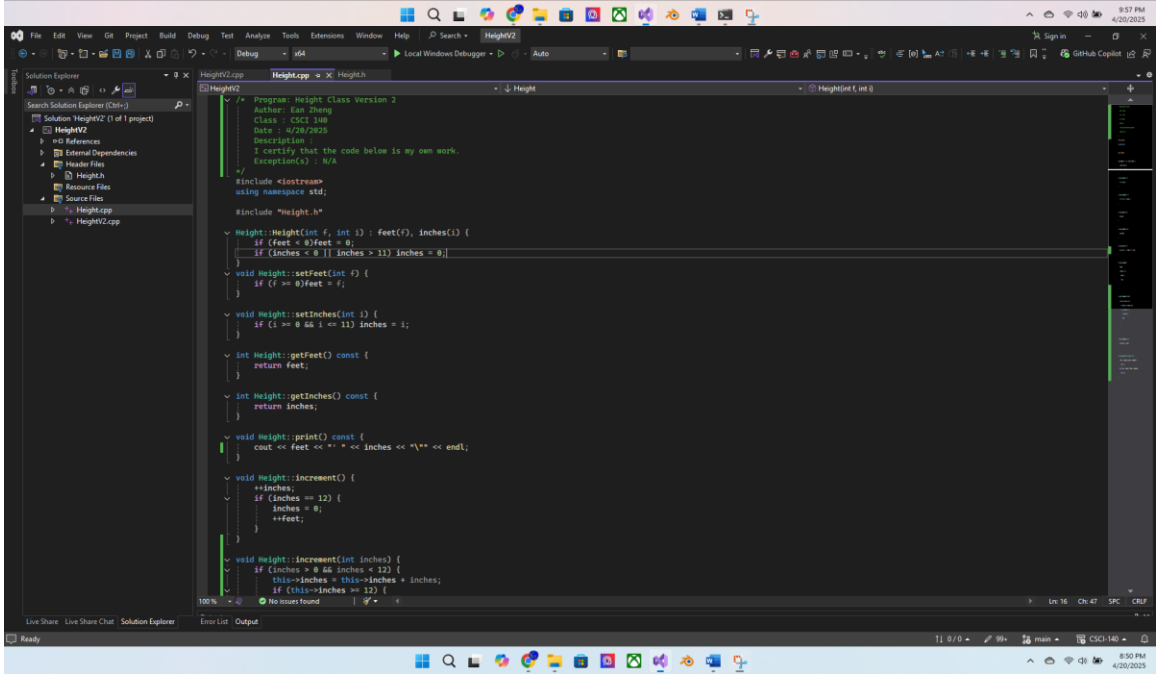
```
/*  Program: Height Class Version 2
   Author: Ean Zheng
   Class : CSCI 140
   Date : 4/20/2025
   Description :
   I certify that the code below is my own work.
   Exception(s) : N/A
*/
#ifndef HEIGHT_H
#define HEIGHT_H

#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
```

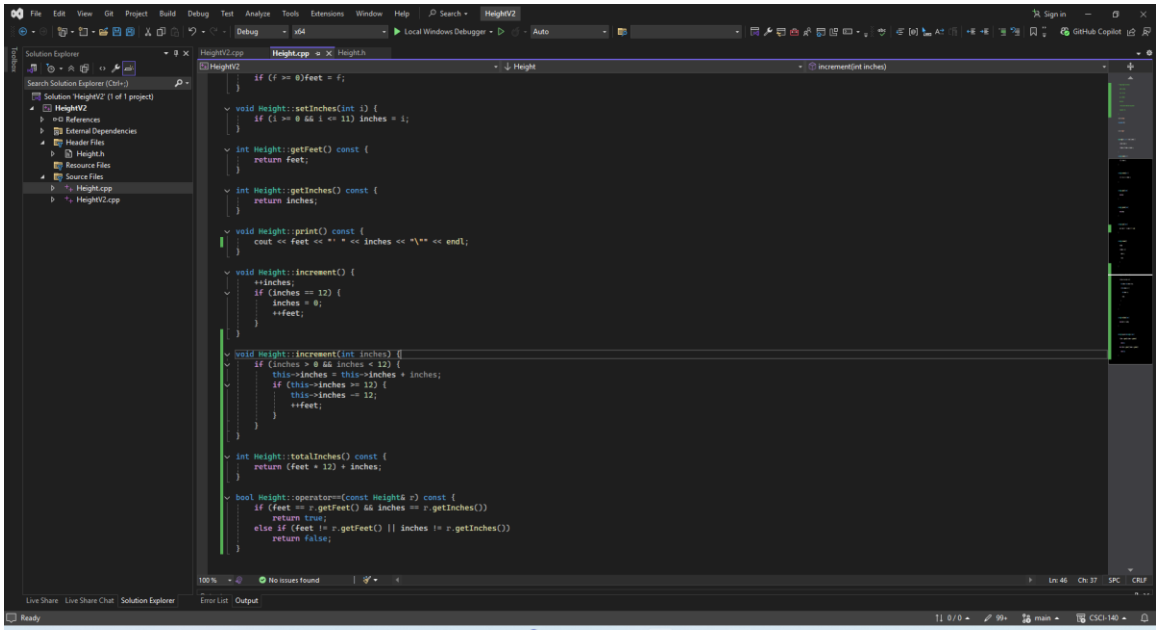


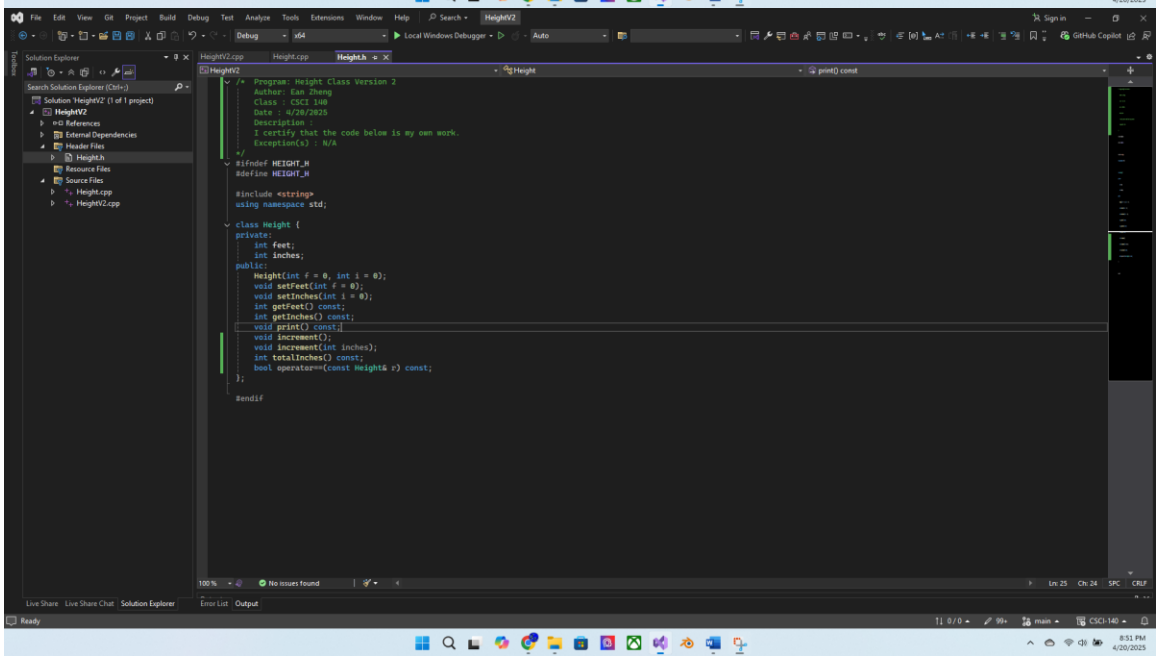


```
    Height h3(5, 8); // feet: 5, inches: 8
    Height h4(-1, 5); // feet: 0, inches: 5 (invalid feet so set to 0)
    Height h5(6, 15); // feet: 6, inches: 0 (invalid inches so set to 0)
    // Print height h3
    Height array[]{ h1, h2, h3, h4, h5 };
    for (int i = 0; i < 5; ++i) {
        array[i].print();
    }
    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
}
```



```
/*  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) {
```



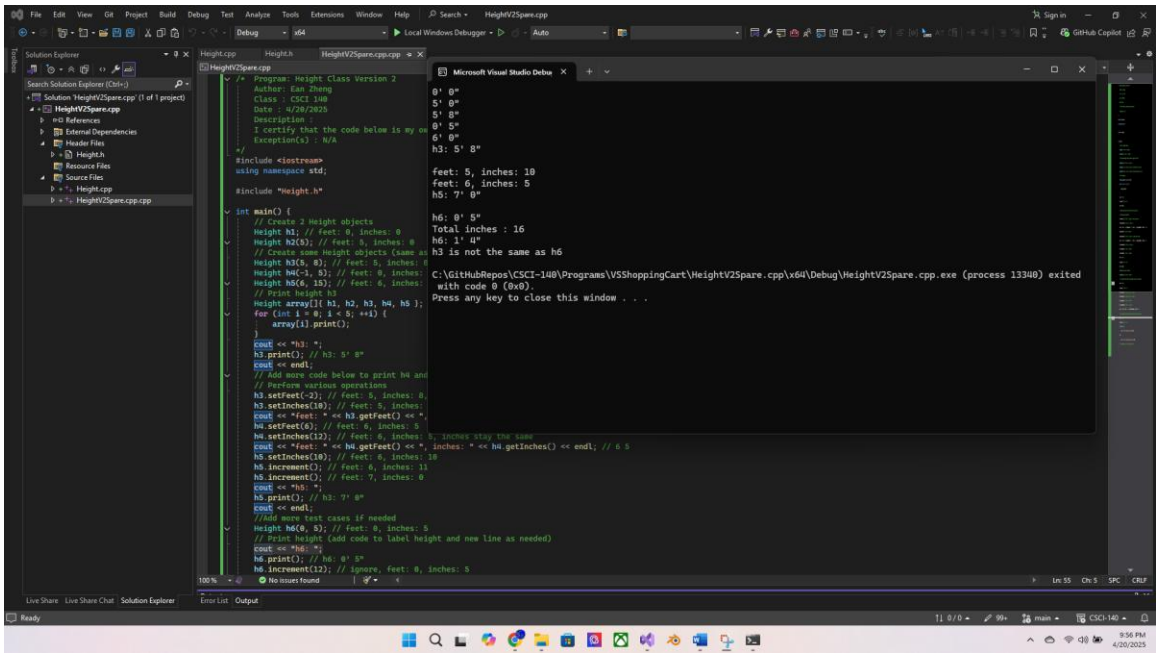




Input/output below:
0' 0"
5' 0"
5' 8"
0' 5"
6' 0"
h3: 5' 8"

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




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"