



COLORADO STATE UNIVERSITY  
— GLOBAL —

## CSC500: PRINCIPLES OF PROGRAMMING

**Credit Hours:** 3

**Contact Hours:** This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

**Faculty Information:** Faculty contact information and office hours can be found on the faculty profile page.

---

### COURSE DESCRIPTION AND OUTCOMES

---

#### Course Description:

In this graduate course, students are provided with a detailed overview of fundamental programming, design, and testing concepts. Students are introduced to programming constructs and learn how to plan and create basic programming applications. Students will develop applications using common programming structures, which include conditional statements, switches, loops, iteration control structures, and arrays.

#### Course Overview:

This course introduces students to the skills needed to plan and create their own programming applications, user interface design skills, object-oriented concepts, and planning tools such as pseudocode and flowcharts. Students also learn to create and manipulate variables, constants, strings, sequential access files, classes, dictionaries, lists, and arrays. Additionally, students will learn how to create programs using recursion, how to plot and visualize data, and how to develop object-oriented programs using inheritance. This course teaches programming concepts using a hands-on approach where students are able to gain mastery of these skills by performing practical exercises in each module and reviewing videos for visual instruction. After the successful completion of this course, students will be able to develop applications that are used in the workplace.

#### Course Learning Outcomes:

1. Explain the terminology used in programming and the tasks performed by a programmer.
2. Develop applications using variables, constants, selection structures, and repetition structures.
3. Implement a solution that uses arrays.
4. Identify constructs for reading and writing of text files in programming.
5. Develop an application using function procedures and string manipulation.

---

### PARTICIPATION & ATTENDANCE

---

Prompt and consistent attendance in your online courses is essential for your success at CSU Global. Failure to verify your attendance within the first 7 days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your Student Success Counselor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

---

## COURSE MATERIALS

---

### Required:

Lysecky, R., Vahid, F., Berrier, J., Chan, C., Nestler, S., Pardoe, I., Siu, R., Sturdivant, R. X., Watts, K., & Miller, B. (n.d.). *CSC500: Principles of programming*. zyBooks. Retrieved from <https://learn.zybooks.com/zybook/CSC500zyBookInteractiveText2020>

Python programming language software. Retrieved from <https://www.python.org/>

### Suggested:

Gaddis, T. (2018). *Starting out with Python* (4th ed). Pearson: New York. eText ISBN 9780134484693, Print 9780134444321

**NOTE:** All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.

---

## COURSE SCHEDULE

---

### Due Dates

The Academic Week at CSU Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and Peer Responses posted by Sunday at 11:59 p.m. MT. Late posts may not be awarded points.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

---

## WEEKLY READING AND ASSIGNMENT DETAILS

---

### Module 1

#### Readings

- Chapter 1 in *CSC500: Principles of Programming*

- Klein, B. (2017). Execute a Python script. Retrieved from [https://www.python-course.eu/python3\\_execute\\_script.php](https://www.python-course.eu/python3_execute_script.php)
- Klein, B. (2017). Input from the keyboard. Retrieved from [https://www.python-course.eu/python3\\_input.php](https://www.python-course.eu/python3_input.php)
- Klein, B. (2017). Structuring with indentation. Retrieved from [https://www.python-course.eu/python3\\_blocks.php](https://www.python-course.eu/python3_blocks.php)
- Klein, B. (2017). The interpreter, an interactive shell. Retrieved from [https://www.python-course.eu/python3\\_interactive.php2](https://www.python-course.eu/python3_interactive.php2)

### **Discussion (25 points)**

#### **Critical Thinking: Title (75 points)**

##### **OPTION #1: Python Program to Add Two Numbers**

Write a Python program to find the addition and subtraction of two numbers.

Ask the user to input two numbers (num1 and num2). Given those two numbers, add them together to find the output. Also, subtract the two numbers to find the output.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

##### **OPTION #2: Python Program to Check for Speed**

Write a Python program to find the multiplication and division of two numbers.

Ask the user to input two numbers (num1 and num2). Given those two numbers, multiply them together to find the output. Also, divide num1/num2 to find the output.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

## **Module 2**

### **Readings**

- Chapter 2 in *CSC500: Principles of Programming*
- Python Software Foundation. (2017). Expressions. Retrieved from <https://docs.python.org/3/reference/expressions.html>
- Singh, M. (2019, June 12). Short circuiting techniques in Python. Retrieved from <https://www.geeksforgeeks.org/short-circuiting-techniques-python/>

### **Discussion (25 points)**

#### **Critical Thinking: Title (75 points)**

##### **OPTION #1: Retail Total**

A customer in a store is purchasing five retail items. Write a program that asks for the price of each item and then displays the subtotal of the sale, the amount of sales tax, and the total. Assume the sales tax is 7 percent.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

#### **OPTION #2: Restaurant Total**

Write a program that calculates the total amount of a meal purchased at a restaurant. The program should ask the user to enter the charge for the food and then calculate the amounts with an 18 percent tip and 7 percent sales tax. Display each of these amounts and the total price.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

### **Module 3**

#### **Readings**

- Chapter 3 in *CSC500: Principles of Programming*
- Aggarwal, N. (2019, November 28). Python data types. Retrieved from <https://www.geeksforgeeks.org/python-data-types/>
- Tiwar, A. (2018, December 13). Python arrays. Retrieved from <https://www.geeksforgeeks.org/python-arrays/>
- Tutorialspoint. (n.d.) NumPy - Iterating over array. Retrieved from [https://www.tutorialspoint.com/numpy/numpy\\_iterating\\_over\\_array](https://www.tutorialspoint.com/numpy/numpy_iterating_over_array)

#### **Discussion (25 points)**

#### **Critical Thinking: Title (75 points)**

##### **OPTION #1: Alarm Time**

Many people keep time using a 24-hour clock (11 is 11am and 23 is 11pm, 0 is midnight). If it is currently 13 and you set your alarm to go off in 50 hours, it will be 15 (3pm). Write a Python program to solve the general version of the above problem. Ask the user for the time now (in hours) and then ask for the number of hours to wait for the alarm. Your program should output what the time will be on a 24-hour clock when the alarm goes off.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

##### **OPTION #2: Sum of Arrays**

Write a Python program to check the sum of three elements from three arrays. Please have the code check to see if the sum is equal to the target value (inputted from the user). Then have the program print the three-element combinations.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

#### **Portfolio Milestone (50 points)**

##### **OPTION #1: Online Shopping Cart**

Step 1: Build the ItemToPurchase class with the following specifications:

- Attributes
- item\_name (string)
- item\_price (float)
- item\_quantity (int)
- Default constructor
- Initializes item's name = "none", item's price = 0, item's quantity = 0
- Method
- print\_item\_cost()

Example of print\_item\_cost() output:

Bottled Water 10 @ \$1 = \$10

Step 2: In the main section of your code, prompt the user for two items and create two objects of the ItemToPurchase class.

Example:

Item 1

Enter the item name:

Chocolate Chips

Enter the item price:

3

Enter the item quantity:

1

Item 2

Enter the item name:

Bottled Water

Enter the item price:

1

Enter the item quantity:

10

(3) Add the costs of the two items together and output the total cost.

Example:

TOTAL COST

Chocolate Chips 1 @ \$3 = \$3

Bottled Water 10 @ \$1 = \$10

Total: \$13

Your program submission materials must include your source code and screenshots of the application executing the code and the results.

### **OPTION #2: Soccer Team Roster**

Step 1: Prompt the user to enter four numbers, each corresponding to a person's weight in pounds. Store all weights in a list. Output the list.

Example:

Enter weight 1:

236.0

Enter weight 2:

89.5

Enter weight 3:

176.0

Enter weight 4:

166.3

Weights: [236.0, 89.5, 176.0, 166.3]

Step 2: Output the average of the list's elements with two digits after the decimal point. Hint: Use a conversion specifier to output with a certain number of digits after the decimal point. Output the max list element with two digits after the decimal point.

Example:

Enter weight 1:

236.0

Enter weight 2:

89.5

Enter weight 3:

176.0

Enter weight 4:

166.3

Weights: [236.0, 89.5, 176.0, 166.3]

Average weight: 166.95

Max weight: 236.00

Your program submission materials must include your source code and screenshots of the application executing the code and the results.

## Module 4

### Readings

- Chapter 4 in *CSC500: Principles of Programming*
- Kaur, H. (2017, June 7). Loops in Python. Retrieved from <https://www.geeksforgeeks.org/loops-in-python/>
- Klein, B. (2017). Python loops. Retrieved from [https://www.python-course.eu/python3\\_loops.php](https://www.python-course.eu/python3_loops.php)
- Sturtz, J. (2019, June 20). Python “while” loops (indefinite iteration). Retrieved from <https://realpython.com/python-while-loop/>
- Tagliaferri, L. (2019, September 19). How to conduct for loops in Python 3. Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-construct-for-loops-in-python-32>

### Discussion (25 points)

#### Critical Thinking: Title (75 points)

##### **OPTION #1: Rainfall**

Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

##### **OPTION #2: Budget**

Write a program that asks the user to enter the amount that they have budgeted for a month. A loop should then prompt the user to enter each of their expenses for the month and keep a running total. When the loop finishes, the program should display the amount that the user is over or under budget.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

## Module 5

### Readings

- Chapter 5 in *CSC500: Principles of Programming*
- Chauhan, P. (2018, September, 11). Python membership and identity operators | in, not in, is, is not. Retrieved from <https://www.geeksforgeeks.org/python-membership-identity-operators-not-not/>
- Harrington, A.N. (n.d.) If statements. Retrieved from <http://anh.cs.luc.edu/handsonPythonTutorial/ifstatements.html>
- Ked, N. (2019, June 12). Statement, indentation, and comment in Python. Retrieved from <https://www.geeksforgeeks.org/statement-indentation-and-comment-in-python/1>

### Discussion (25 points)

### Mastery Exercise (10 points)

### Critical Thinking: Title (75 points)

#### **OPTION #1: Mixing Colors**

The colors red, blue, and yellow are known as the primary colors because they cannot be made by mixing other colors. When you mix two primary colors, you get a secondary color, as shown here:

- When you mix red and blue, you get purple.
- When you mix red and yellow, you get orange.
- When you mix blue and yellow, you get green.

Design a program that prompts the user to enter the names of two primary colors to mix. If the user enters anything other than “red,” “blue,” or “yellow,” the program should display an error message. Otherwise, the program should display the name of the secondary color that results.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

#### **OPTION #2: Book Club**

The CSU Global Bookstore has a book club that awards points to its students based on the number of books purchased each month. The points are awarded as follows:

- If a customer purchases 0 books, they earn 0 points.
- If a customer purchases 2 books, they earn 5 points.
- If a customer purchases 4 books, they earn 15 points.
- If a customer purchases 6 books, they earn 30 points.

- If a customer purchases 8 or more books, they earn 60 points.

Write a program that asks the user to enter the number of books that they have purchased this month and then display the number of points awarded.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

## Module 6

### Readings

- Chapter 6 in *CSC500: Principles of Programming*
- Cánepa, G. (2018, November 25). Manipulating lists and dictionaries in Python. Retrieved from <https://www.pluralsight.com/guides/manipulating-lists-dictionaries-python>
- Klein, B. (2017). Dictionaries. Retrieved from [https://www.python-course.eu/python3\\_dictionaries.php](https://www.python-course.eu/python3_dictionaries.php)
- Klein, B. (2017). Lists. Retrieved from [https://www.python-course.eu/python3\\_list\\_manipulation.php1](https://www.python-course.eu/python3_list_manipulation.php1)

### Discussion (25 points)

#### Critical Thinking: Title (75 points)

##### **OPTION #1: Driver's License Exam**

The local driver's license office has asked you to create an application that grades the written portion of the driver's license exam. The exam has 20 multiple-choice questions. Here are the correct answers:

1. A
2. C
3. A
4. A
5. D
6. B
7. C
8. A
9. C
10. B
11. A
12. D
13. C
14. A
15. D
16. C
17. B
18. B

- 19. D
- 20. A

Your program should store these correct answers in a list. The program should read the student's answers for each of the 20 questions from a text file and store the answers in another list. (Create your own text file to test the application.) After the student's answers have been read from the file, the program should display a message indicating whether the student passed or failed the exam. (A student must correctly answer 15 of the 20 questions to pass the exam.) It should then display the total number of correctly answered questions, the total number of incorrectly answered questions, and a list showing the question numbers of the incorrectly answered questions.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

**OPTION #2: Course Information**

Write a program that creates a dictionary containing course numbers and the room numbers of the rooms where the courses meet. The dictionary should have the following key-value pairs:

<b>Course Number (key)</b>	<b>Room Number (value)</b>
CSC101	3004
CSC102	4501
CSC103	6755
NET110	1244
COM241	1411

The program should also create a dictionary containing course numbers and the names of the instructors that teach each course. The dictionary should have the following key-value pairs:

<b>Course Number (key)</b>	<b>Instructor (value)</b>
CSC101	Haynes
CSC102	Alvarado
CSC103	Rich

Course Number (key)	Instructor (value)
NET110	Burke
COM241	Lee

The program should also create a dictionary containing course numbers and the meeting times of each course. The dictionary should have the following key-value pairs:

Course Number (key)	Meeting Time (value)
CSC101	8:00 a.m.
CSC102	9:00 a.m.
CSC103	10:00 a.m.
NET110	11:00 a.m.
COM241	1:00 p.m.

The program should let the user enter a course number and then it should display the course's room number, instructor, and meeting time.

Compile and submit your source code and screenshots of the application executing the code and the results in a single document.

### **Portfolio Milestone (50 points)**

#### **OPTION #1: Online Shopping Cart**

(4) Build the ShoppingCart class with the following data attributes and related methods. Note: Some can be method stubs (empty methods) initially, to be completed in later steps.

- Parameterized constructor, which takes the customer name and date as parameters
- Attributes
- customer\_name (string) - Initialized in default constructor to "none"
- current\_date (string) - Initialized in default constructor to "January 1, 2020"
- cart\_items (list)
- Methods

- `add_item()`
  - Adds an item to `cart_items` list. Has parameter `ItemToPurchase`. Does not return anything.
- `remove_item()`
  - Removes item from `cart_items` list. Has a string (an item's name) parameter. Does not return anything.
  - If item name cannot be found, output this message: Item not found in cart. Nothing removed.
- `modify_item()`
  - 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, modify item in cart.
  - If item cannot be found (by name) in cart, output this message: Item not found in cart. Nothing modified.
- `get_num_items_in_cart()`
  - Returns quantity of all items in cart. Has no parameters.
- `get_cost_of_cart()`
  - Determines and returns the total cost of items in cart. Has no parameters.
- `print_total()`
  - Outputs total of objects in cart.
  - If cart is empty, output this message: SHOPPING CART IS EMPTY
- `print_descriptions()`
  - Outputs each item's description.

Example of `print_total()` output:

John Doe's Shopping Cart - February 1, 2020

Number of Items: 8

Nike Romaleos 2 @ \$189 = \$378

Chocolate Chips 5 @ \$3 = \$15

Powerbeats 2 Headphones 1 @ \$128 = \$128

Total: \$521

Example of `print_descriptions()` output:

John Doe's Shopping Cart - February 1, 2020

Item Descriptions

Nike Romaleos: Volt color, Weightlifting shoes

Chocolate Chips: Semi-sweet

Powerbeats 2 Headphones: Bluetooth headphones

(5) In the main section of your code, implement the `print_menu()` function. `print_menu()` has a `ShoppingCart` parameter and outputs a menu of options to manipulate the shopping cart. Each option is represented by a single character. Build and output the menu within the function.

If an invalid character is entered, continue to prompt for a valid choice. Hint: Implement `Quit` before implementing other options. Call `print_menu()` in the `main()` function. Continue to execute the menu until the user enters `q` to `Quit`.

Example:

MENU

a - Add item to cart

r - Remove item from cart

c - Change item quantity

i - Output items' descriptions

o - Output shopping cart

q - Quit

Choose an option:

(6) Implement `Output shopping cart` menu option. Implement `Output item's description` menu option.

Example of shopping cart menu option:

OUTPUT SHOPPING CART

John Doe's Shopping Cart - February 1, 2020

Number of Items: 8

Nike Romaleos 2 @ \$189 = \$378

Chocolate Chips 5 @ \$3 = \$15

Powerbeats 2 Headphones 1 @ \$128 = \$128

Total: \$521

Example of item description menu option.

OUTPUT ITEMS' DESCRIPTIONS

John Doe's Shopping Cart - February 1, 2020

## Item Descriptions

Nike Romaleos: Volt color, Weightlifting shoes

Chocolate Chips: Semi-sweet

Powerbeats 2 Headphones: Bluetooth headphones

Your program submission materials must include your source code and screenshots of the application executing the code and the results.

### **OPTION #2:**

(3) Prompt the user to input five pairs of numbers: A player's jersey number (0 - 99) and the player's rating (1 - 9). Store the jersey numbers and the ratings in a dictionary. Output the dictionary's elements with the jersey numbers in ascending order (i.e., output the roster from smallest to largest jersey number). Hint: Dictionary keys can be stored in a sorted list. (3 pts)

Example:

Enter player 1's jersey number:

84

Enter player 1's rating:

7

Enter player 2's jersey number:

23

Enter player 2's rating:

4

Enter player 3's jersey number:

4

Enter player 3's rating:

5

Enter player 4's jersey number:

30

Enter player 4's rating:

2

Enter player 5's jersey number:

66

Enter player 5's rating:

9

ROSTER

Jersey number: 4, Rating: 5

Jersey number: 23, Rating: 4

Jersey number: 30, Rating: 2

...

(4) Implement a menu of options for a user to modify the roster. Each option is represented by a single character. The program initially outputs the menu after a user chooses an option. The program ends when the user chooses the option to Quit. For this step, the other options do nothing.

Example:

MENU

a - Add player

d - Remove player

u - Update player rating

r - Output players above a rating

o - Output roster

q - Quit

Choose an option:

Your program submission materials must include your source code and screenshots of the application executing the code and the results.

## Module 7

### Readings

- Chapter 7 in *CSC500: Principles of Programming*
- Badugu, P. (2018, September 11). Functions in Python. Retrieved from <https://www.geeksforgeeks.org/functions-in-python/>
- Prah, T. N. (2019, November 12). Python input, Python output and import. <https://towardsdatascience.com/python-input-output-and-import-73e875516694?gi=1adbfce9a093>
- Tutorialspoint. (n.d.). Python - Exceptions handling. Retrieved from [https://www.tutorialspoint.com/python/python\\_exceptions.htm](https://www.tutorialspoint.com/python/python_exceptions.htm)

### Discussion (25 points)

## Module 8

### Readings

- Chapter 8 in *CSC500: Principles of Programming*
- Geeks for Geeks. (2019, October 1). Python string. Retrieved from <https://www.geeksforgeeks.org/python-strings/>

- Python Software Foundation (2017). Classes. Retrieved from <https://docs.python.org/3/tutorial/classes.html>

### Discussion (25 points)

### Portfolio Project (250 points)

#### **OPTION #1: Online Shopping Cart**

From Milestone 1:

Step 1: Build the ItemToPurchase class with the following specifications:

- Attributes
- item\_name (string)
- item\_price (float)
- item\_quantity (int)
- Default constructor
- Initializes item's name = "none", item's price = 0, item's quantity = 0
- Method
- print\_item\_cost()

Example of print\_item\_cost() output:

Bottled Water 10 @ \$1 = \$10

Step 2: In the main section of your code, prompt the user for two items and create two objects of the ItemToPurchase class.

Example:

Item 1

Enter the item name:

Chocolate Chips

Enter the item price:

3

Enter the item quantity:

1

Item 2

Enter the item name:

Bottled Water

Enter the item price:

1

Enter the item quantity:

Step 3: Add the costs of the two items together and output the total cost.

Example:

TOTAL COST

Chocolate Chips 1 @ \$3 = \$3

Bottled Water 10 @ \$1 = \$10

Total: \$13

Fix any issues from milestone 1 submission prior to submitting the Portfolio Project.

From Milestone 2:

Step 4: Build the ShoppingCart class with the following data attributes and related methods. Note: Some can be method stubs (empty methods) initially, to be completed in later steps.

- Parameterized constructor, which takes the customer name and date as parameters
- Attributes
- customer\_name (string) - Initialized in default constructor to "none"
- current\_date (string) - Initialized in default constructor to "January 1, 2020"
- cart\_items (list)
- Methods
- add\_item()
  - Adds an item to cart\_items list. Has parameter ItemToPurchase. Does not return anything.
- remove\_item()
  - Removes item from cart\_items list. Has a string (an item's name) parameter. Does not return anything.
  - If item name cannot be found, output this message: Item not found in cart. Nothing removed.
- modify\_item()
  - 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, modify item in cart.
  - If item cannot be found (by name) in cart, output this message: Item not found in cart. Nothing modified.
- get\_num\_items\_in\_cart()

- Returns quantity of all items in cart. Has no parameters.
- `get_cost_of_cart()`
  - Determines and returns the total cost of items in cart. Has no parameters.
- `print_total()`
  - Outputs total of objects in cart.
  - If cart is empty, output this message: SHOPPING CART IS EMPTY
- `print_descriptions()`
  - Outputs each item's description.

Example of `print_total()` output:

John Doe's Shopping Cart - February 1, 2020

Number of Items: 8

Nike Romaleos 2 @ \$189 = \$378

Chocolate Chips 5 @ \$3 = \$15

Powerbeats 2 Headphones 1 @ \$128 = \$128

Total: \$521

Example of `print_descriptions()` output:

John Doe's Shopping Cart - February 1, 2020

Item Descriptions

Nike Romaleos: Volt color, Weightlifting shoes

Chocolate Chips: Semi-sweet

Powerbeats 2 Headphones: Bluetooth headphones

Step 5: In the main section of your code, implement the `print_menu()` function. `print_menu()` has a `ShoppingCart` parameter and outputs a menu of options to manipulate the shopping cart. Each option is represented by a single character. Build and output the menu within the function.

If an invalid character is entered, continue to prompt for a valid choice. Hint: Implement Quit before implementing other options. Call `print_menu()` in the `main()` function. Continue to execute the menu until the user enters q to Quit.

Example:

MENU

a - Add item to cart

r - Remove item from cart

c - Change item quantity

i - Output items' descriptions

o - Output shopping cart

q - Quit

Choose an option:

Step 6: Implement Output shopping cart menu option. Implement Output item's description menu option.

Example of shopping cart menu option:

OUTPUT SHOPPING CART

John Doe's Shopping Cart - February 1, 2020

Number of Items: 8

Nike Romaleos 2 @ \$189 = \$378

Chocolate Chips 5 @ \$3 = \$15

Powerbeats 2 Headphones 1 @ \$128 = \$128

Total: \$521

Example of item description menu option.

OUTPUT ITEMS' DESCRIPTIONS

John Doe's Shopping Cart - February 1, 2020

Item Descriptions

Nike Romaleos: Volt color, Weightlifting shoes

Chocolate Chips: Semi-sweet

Powerbeats 2 Headphones: Bluetooth headphones

Fix any issues from milestone 2 submission prior to submitting the Portfolio Project.

Additional tasks for the final project submission:

Step 7: In the main section of your code, prompt the user for a customer's name and today's date. Output the name and date. Create an object of type ShoppingCart.

Example:

Enter customer's name:

John Doe

Enter today's date:

February 1, 2020

Customer name: John Doe

Today's date: February 1, 2020

Step 8: Implement Add item to cart menu option.

Example:

ADD ITEM TO CART

Enter the item name:

Nike Romaleos

Enter the item description:

Volt color, Weightlifting shoes

Enter the item price:

189

Enter the item quantity:

2

Step 9: Implement remove item menu option.

Example:

REMOVE ITEM FROM CART

Enter name of item to remove:

Chocolate Chips

Step 10: Implement Change item quantity menu option. Hint: Make new ItemToPurchase object before using ModifyItem() method.

Example:

CHANGE ITEM QUANTITY

Enter the item name:

Nike Romaleos

Enter the new quantity:

3

Your final program submission materials must include your source code and screenshots of the application executing the code and the results.

### **OPTION #2: Soccer Team Roster**

This program will store roster and rating information for a soccer team. Coaches rate players during tryouts to ensure a balanced team.

From Milestone 1:

Step 1: Prompt the user to enter four numbers, each corresponding to a person's weight in pounds. Store all weights in a list. Output the list.

Example:

Enter weight 1:

236.0

Enter weight 2:

89.5

Enter weight 3:

176.0

Enter weight 4:

166.3

Weights: [236.0, 89.5, 176.0, 166.3]

Step 2: Output the average of the list's elements with two digits after the decimal point. Hint: Use a conversion specifier to output with a certain number of digits after the decimal point. Output the max list element with two digits after the decimal point.

Example:

Enter weight 1:

236.0

Enter weight 2:

89.5

Enter weight 3:

176.0

Enter weight 4:

166.3

Weights: [236.0, 89.5, 176.0, 166.3]

Average weight: 166.95

Max weight: 236.00

Fix any issues with milestone 1 submission prior to submitting the Portfolio Project.

From milestone 2:

Step 3: Prompt the user to input five pairs of numbers: A player's jersey number (0 - 99) and the player's rating (1 - 9). Store the jersey numbers and the ratings in a dictionary. Output the dictionary's elements with the jersey numbers in ascending order (i.e., output the roster from smallest to largest jersey number). Hint: Dictionary keys can be stored in a sorted list.

Example:

Enter player 1's jersey number:

84

Enter player 1's rating:

7

Enter player 2's jersey number:

23

Enter player 2's rating:

4

Enter player 3's jersey number:

4

Enter player 3's rating:

5

Enter player 4's jersey number:

30

Enter player 4's rating:

2

Enter player 5's jersey number:

66

Enter player 5's rating:

9

ROSTER

Jersey number: 4, Rating: 5

Jersey number: 23, Rating: 4

Jersey number: 30, Rating: 2

...

Step 4: Implement a menu of options for a user to modify the roster. Each option is represented by a single character. The program initially outputs the menu after a user chooses an option. The program ends when the user chooses the option to Quit. For this step, the other options do nothing.

Example:

MENU

a - Add player

d - Remove player

u - Update player rating

r - Output players above a rating

o - Output roster

q - Quit

Choose an option:

Fix any issues from milestone 2 submission prior to submitting the Portfolio Project.

Additional tasks to add before submitting the final project:

Step 5: Implement the "Add player" menu option. Prompt the user for a new player's jersey number and rating. Append the values to the two vectors.

Example:

Enter a new player's jersey number:

49

Enter the player's rating:

8

Step 6: Implement the "Delete player" menu option. Prompt the user for a player's jersey number. Remove the player from the roster (delete the jersey number and rating).

Example:

Enter a jersey number:

4

Step 7: Implement the "Output players above a rating" menu option. Prompt the user for a rating. Print the jersey number and rating for all players with ratings above the entered value.

Example:

Enter a rating:

5

ABOVE 5

Jersey number: 66, Rating: 9

Jersey number: 84, Rating: 7

...

Your final program submission materials must include your source code and screenshots of the application executing the code and the results.

---

## COURSE POLICIES

---

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

### Course Grading

20% Discussion Participation  
45% Critical Thinking Assignments  
35% Final Portfolio Project

---

## IN-CLASSROOM POLICIES

---

For information on late work and incomplete grade policies, please refer to our [In-Classroom Student Policies and Guidelines](#) or the Academic Catalog for comprehensive documentation of CSU Global institutional policies.

### **Academic Integrity**

Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /re-purposing your own work (see *CSU Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and internet resources.

### **Citing Sources with APA Style**

All students are expected to follow the *CSU Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on APA style, please review the APA resources within the CSU Global Library under the “APA Guide & Resources” link. A link to this document should also be provided within most assignment descriptions in your course.

### **Disability Services Statement**

CSU Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email [ada@CSUGlobal.edu](mailto:ada@CSUGlobal.edu) for additional information to coordinate reasonable accommodations for students with documented disabilities.

### **Netiquette**

Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults, or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments relating to ethnicity, religion, age, disability, sex (or sexual orientation), or race, in addition to unwanted sexual advances or intimidations by email or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.