

## Team Project

### Textbook

## Team Project

*Note: This project create's code that can be used in an app, desktop application, or website.*



Most programming projects in the real world are not built by one person. They are often built by many people working on different smaller parts of the program to create something large and more complicated. They make sure their particular segment of code works well with the group and it produces reliable outputs.

We will get the chance to practice working on a project with a small team to create something more complex.

## Receipt Project

As a group, you will work together to build a receipt that you might get at the store.

This project is designed to be done in a group of 3 people. It is **not autograded** so that you and your team are free to create something incredible without restraint. See what you can create with what you have learned in this course so far!

- One person is in charge of finding the **total** price.

- One person is in charge of figuring out **tax**.
- One person is in charge of figuring out **change**.

## All Together Example Output

Here is an example of the finished product. Your code should print out something similar to the following:

```
['milk', 4, 'cheese', 6, 'apples', 4, 'cereal', 8, 'carrots', 2, 'shoes', 30, 'soap', 6, 'garbage bag', 14, 'toilet paper', 16, 'laundry detergent', 20]
```

```
Your total price before tax is $110
```

```
Total for food: $24
```

```
Total for merchandise: $86
```

```
Food tax total: $0.72
```

```
Merchandise tax total: $4.3
```

```
Your grand total is: $115.02
```

```
Your change is 384.98 dollars
```

```
You will get 3 hundred dollar bills back.
```

```
You will get 8 ten dollar bills back.
```

```
You will get 4 one dollar bills back.
```

```
You will get 0.98 in coins back.
```

Let's look at some further instructions for each role.

## Total Price

The store you will be shopping at has the following items with their prices:

Item	Price (in dollars)
milk	2
shoes	3
soap	2
cheese	4
apples	1
garbage bags	15
cereal	3
carrots	7
toilet paper	8

laundry detergent	10
-------------------	----

Your task is to ask the user in an input statement how many of each item they want to buy. Then, tally up the total cost the user owes.

Print a total list of each item. This list should have the total for each item after it depending on how many of each item the user wants.

Here is an example of the printed list assuming the user wanted 2 of each item.

```
['milk', 4, 'cheese', 6, 'apples', 4, 'cereal', 8, 'carrots', 2, 'shoes', 30, 'soap', 6, 'garbage bag', 14, 'toilet paper', 16, 'laundry detergent', 20]
```

Print out the total price before tax like this:

```
Your total price before tax is $110
```

## Tax

Did you know that sales tax is different for food than it is for other items? Your task is to calculate how much tax would be, depending on what the user chooses to buy.

For this assignment, assume that **tax for food items is 3%** and **tax for other items is 5%**.

You don't have to wait for your teammate to finish creating the code to tally up the total to get started. You can start at the same time. Just use a pretend number as a total as a substitute until you can get the real number from your teammate.

Work with your teammate who is in charge of tallying up totals to make sure they separate the food total from the other merchandise. Explain to your teammate that you need it separated in such a way so that you can find the individual totals for food and for merchandise.

Here is an example output for the tax section.

```
Your total price before tax is $110
```

```
Total for food: $24
```

```
Total for merchandise: $86
```

```
Food tax total: $0.72
```

```
Merchandise tax total: $4.3
```

```
Your grand total is: $115.02
```

## Change

Your task is to calculate how much change the user would receive. Inform the user in an alert statement what the total is, and ask if they want to pay using a 10, 50, 100, or 500 dollar bill. If the bill they chose wasn't big enough, let the user know they need to use a bigger bill and to start over. (Don't over complicate this step—it's okay to make the user start over)

You don't have to wait for your teammate to finish creating the code to tally up the total to get started. You can start at the same time. Just use a pretend number as a total as a substitute until you can get the real number from your teammate.

Keep in mind that you will need to calculate how much change they get after paying the total AFTER tax. This will likely have some decimals involved.

Let the user know how many hundred dollar bills, ten dollar bills, one dollar bills, and amounts in coins they will get back.

Here is an example output:

```
Your grand total is: $115.02
```

Your change is 384.98 dollars

You will get 3 hundred dollar bills back.

You will get 8 ten dollar bills back.

You will get 4 one dollar bills back.

You will get 0.98 in coins back.

Hint: You will need to use the modulus to get the individual number of bills.

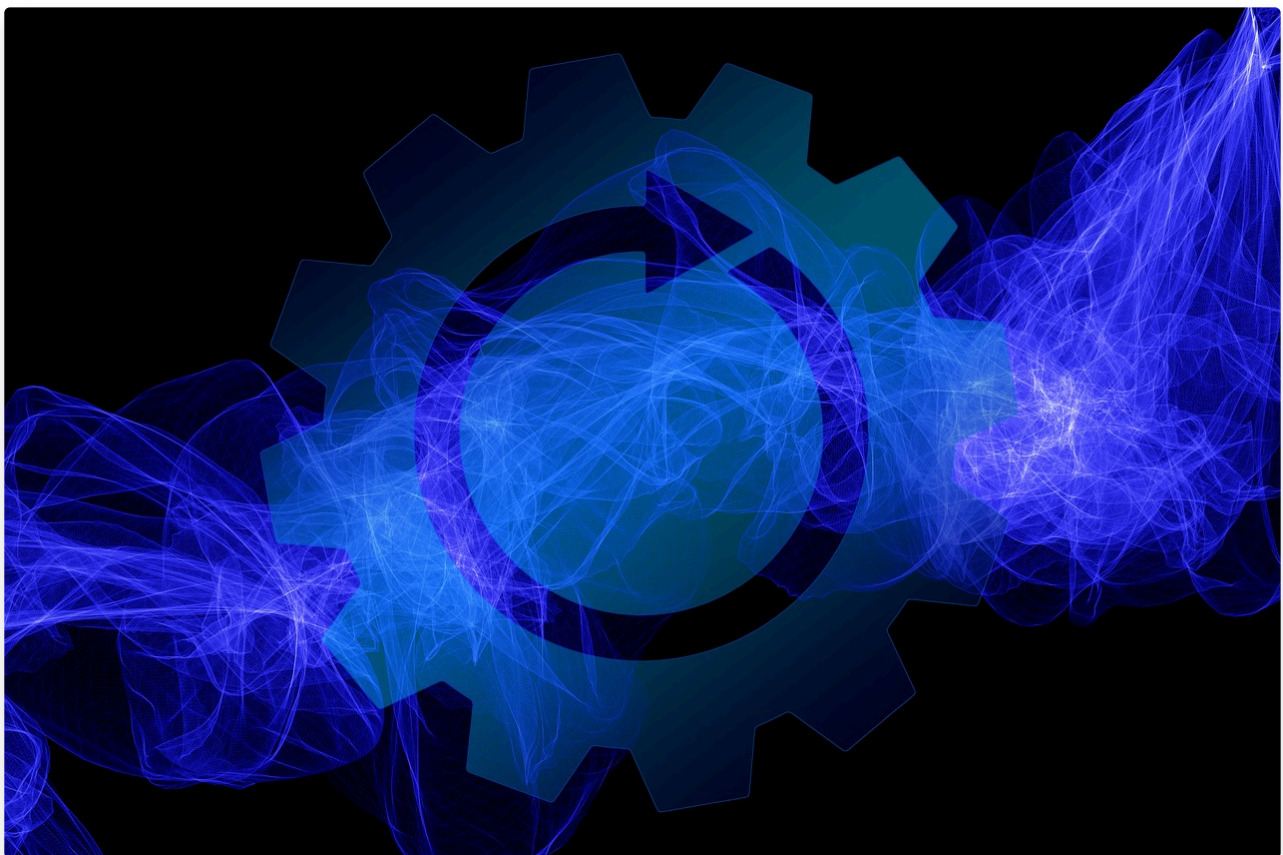
Hint: If you change a float to an integer, it drops the decimal value.

## Tips to Get Started

You can work on your individual Skill Struck accounts and join your code together at the end to make sure it all works together. You don't need to wait for each other to get started—just use some example numbers until you can get the real values from your teammates.

We also recommend that you follow steps in the development life cycle.

## Software Development Life Cycle (SDLC)



The Software Development Life Cycle is a method of working on software projects in a way that helps to make quality programs. It consists of the following steps.

1. Planning – Decide what your project will do and how to get there
2. Define Requirements – Get the requirements very clear
3. Design the Program – brainstorm, plan, storyboard, diagram, and layout the program
4. Software Development – Build the program
5. Testing – Test to see if your program works with a variety of circumstances



6. Operations and Maintenance – Keep your program updated and clear out bugs that come up

Following these steps will help guide you as you approach your group project.

## Joining Code Together

When you come together to join your code, take some time as a team to understand each other's code. You will find it helpful to write clean code so that others can easily understand your code when they read it. Here are some tips to writing clean code:

- name your variables something that makes sense for its purpose
- add comments throughout the code explaining what each code segment does

You may need to tweak some variable names to make sure you don't accidentally duplicate variable names. Make sure you are using the right variables in the right place when you join your code together.

Give each other some time to explain their code segments and how they work before you try to join code together. Allow them to explain why they designed their code the way they did and discuss what strengths and weaknesses different approaches might have.

## Code Review

As you join code together, take some time to conduct a code review. Each team member should review their other two team member's code.

A code review is where you carefully look through someone else's code. The purpose of a code review is to help make sure the code does what it is designed to do.

Look for strengths and weaknesses. Check for any bugs or flaws in logic. Make sure it will work well with other team members.

Be sure to use the code review as an opportunity to point out strengths in the code as well! Help build your team members up and celebrate what you have learned so far!

## Feedback



After you have joined your code together and it's working smoothly, look beyond your team for feedback. It's important to gather ideas and opinions from a variety of people with different backgrounds in order to minimize possible bias in your program. Asking for feedback will also allow people with all abilities to try your program and see if it works for them as well.

Programmers in the industry ask for feedback from thousands of users from around the world in order to make a really solid program. The more input they get from more diverse users, the better their project will be. We will just look for feedback from your classmates for now.

Show your program to other teams in your class. Take notes on their user experience. What goes well? What could be improved? Where were any bugs that surfaced? Did any certain users have a hard time with the program? What patterns emerged? Make sure to give feedback to at least 3 other teams and get feedback from 3 other teams.

### **Don't forget to apply the feedback!**

Take your notes that you gathered back to your team and make improvements where you can!

This will help you polish your group project until it shines and is usable by more people!

## **Present your Project**

Work together with your team to show off your project! A smooth presentation really shows off all the hard work you have done! Imagine if a large tech company developed a shiny new piece of tech. At the big company meeting where they introduce the project they weren't practiced and fumble through the presentation. It would reflect badly on their project.

Practice your presentation with your team so that you can show off your project in a way that makes it shine!

## **Challenges (3)**

### **1. Person #1: Calculate the Total**

Your task is to ask the user in an input statement how many of each item they want to buy. Then, tally up the total cost the user owes.

Print a total list of each item. This list should have the total for each item after it depending on how many of each item the user wants.

Here is an example of the printed list assuming the user wanted 2 of each item.

```
['milk', 4, 'cheese', 6, 'apples', 4, 'cereal', 8, 'carrots', 2, 'shoes', 30, 'soap', 6, 'garbage bag', 14, 'toilet paper', 16, 'laundry detergent', 20]
```

Print out the total price before tax like this:

```
Your total price before tax is $110
```

#### **Requirements:**

- These are not autograded -- you will need to visit with your teacher about this project. You can submit it as an assignment, but it will not be graded by Skill Struck.

## 2. Person #2: Calculating Tax

Did you know that sales tax is different for food than it is for other items? Your task is to calculate how much tax would be, depending on what the user chooses to buy.

For this assignment, assume that **tax for food items is 3%** and **tax for other items is 5%**.

You don't have to wait for your teammate to finish creating the code to tally up the total to get started. You can start at the same time. Just use a pretend number as a total as a substitute until you can get the real number from your teammate.

Work with your teammate who is in charge of tallying up totals to make sure they separate the food total from the other merchandise. Explain to your teammate that you need it separated in such a way so that you can find the individual totals for food and for merchandise.

Here is an example output for the tax section.

```
Your total price before tax is $110
```

```
Total for food: $24
```

```
Total for merchandise: $86
```

```
Food tax total: $0.72
```

```
Merchandise tax total: $4.3
```

```
Your grand total is: $115.02
```

### Requirements:

- These are not autograded -- you will need to visit with your teacher about this project. You can submit it as an assignment, but it will not be graded by Skill Struck.

## 3. Person #3: Calculating Change

Your task is to calculate how much change the user would receive. Inform the user in an alert statement what the total is, and ask if they want to pay using a 10, 50, 100, or 500 dollar bill. If the bill they chose wasn't big enough, let the user know they need to use a bigger bill and to start over. (Don't over complicate this step—it's okay to make the user start over)

You don't have to wait for your teammate to finish creating the code to tally up the total to get started. You can start at the same time. Just use a pretend number as a total as a substitute until you can get the real number from your teammate.

Keep in mind that you will need to calculate how much change they get after paying the total AFTER tax. This will likely have some decimals involved.

Let the user know how many hundred dollar bills, ten dollar bills, one dollar bills, and amounts in coins they will get back.

### Requirements:

- These are not autograded -- you will need to visit with your teacher about this project. You can submit it as an assignment, but it will not be graded by Skill Struck.

---

## Answer Keys & Solutions

---

### Challenges

---

#### 1. Person #1: Calculate the Total

Solution:

```
1 #CALCULATING THE TOTAL
2 receipt = []
3 prices = []
4 food_prices = []
5 merchandise_prices = []
6 def response(item, price):
7     answer = input("How many of this item do you want? " + item)
8     receipt.append(item)
9     item_cost = price * int(answer)
10    receipt.append(item_cost)
11    prices.append(item_cost)
12    if item == "milk":
13        food_prices.append(item_cost)
14    elif item == "cheese":
15        food_prices.append(item_cost)
16    elif item == "apples":
17        food_prices.append(item_cost)
18    elif item == "cereal":
19        food_prices.append(item_cost)
20    elif item == "carrots":
21        food_prices.append(item_cost)
22    else:
23        merchandise_prices.append(item_cost)
24
25 response("milk", 2)
26 response("cheese", 3)
27 response("apples", 2)
28 response("cereal", 4)
29 response("carrots", 1)
30 response("shoes", 15)
31 response("soap", 3)
32 response("garbage bag", 7)
33 response("toilet paper", 8)
34 response("laundry detergent", 10)
35
36 total = 0
37 for x in prices:
38     total = total + x
39
40 print(receipt)
41
```



```
42 print("Your total price before tax is $" + str(total))
```

## 2. Person #2: Calculating Tax

Solution:

```
1 # Calculating Tax
2
3 food_total = 0
4 for x in food_prices:
5     food_total = food_total + x
6
7 merchandise_total = 0
8 for x in merchandise_prices:
9     merchandise_total = merchandise_total + x
10
11 print("Total for food: $" + str(food_total))
12 print("Total for merchandise: $" + str(merchandise_total))
13
14 food_tax = food_total * .03
15 food_tax = round(food_tax, 2)
16 sales_tax = merchandise_total * .05
17 sales_tax = round(sales_tax, 2)
18
19
20 print("Food tax total: $" + str(food_tax))
21 print("Merchandise tax total: $" + str(sales_tax))
22
23 total = total + food_tax + sales_tax
24
25 print("Your grand total is: $" + str(total))
```

## 3. Person #3: Calculating Change

Solution:

```
1 bills = int(input("Your total is: " + str(total) + " dollars. What would you like to
  pay with? A 10, 50, 100, or 500 dollar bill? If your total is too high, start again
  and get fewer groceries.))
2
3 if bills == 10:
4     change = 10 - total
5     print("Your change is " + str(change) + " dollars")
6     coins = change % 1
7     coins = round(coins, 2)
8     ones = change % 10
9     ones = int(ones)
10    tens = change % 100
11    tens = tens / 10
12    tens = int(tens)
13    hundreds = change % 1000
14    hundreds = hundreds/100
15    hundreds = int(hundreds)
16    print("You will get " + str(hundreds) + " hundreds dollar bills back.")
17    print("You will get " + str(tens) + " ten dollar bills back.")
```

```
18     print("You will get " + str(ones) + " one dollar bills back.")
19     print("You will get " + str(coins) + " in coins back.")
20 elif bills == 50:
21     change = 50 - total
22     print("Your change is " + str(change) + " dollars")
23     coins = change % 1
24     coins = round(coins, 2)
25     ones = change % 10
26     ones = int(ones)
27     tens = change % 100
28     tens = tens / 10
29     tens = int(tens)
30     hundreds = change % 1000
31     hundreds = hundreds/100
32     hundreds = int(hundreds)
33     print("You will get " + str(hundreds) + " hundreds dollar bills back.")
34     print("You will get " + str(tens) + " ten dollar bills back.")
35     print("You will get " + str(ones) + " one dollar bills back.")
36     print("You will get " + str(coins) + " in coins back.")
37 elif bills == 100:
38     change = 100 - total
39     print("Your change is " + str(change) + " dollars")
40     coins = change % 1
41     coins = round(coins, 2)
42     ones = change % 10
43     ones = int(ones)
44     tens = change % 100
45     tens = tens / 10
46     tens = int(tens)
47     hundreds = change % 1000
48     hundreds = hundreds/100
49     hundreds = int(hundreds)
50     print("You will get " + str(hundreds) + " hundreds dollar bills back.")
51     print("You will get " + str(tens) + " ten dollar bills back.")
52     print("You will get " + str(ones) + " one dollar bills back.")
53     print("You will get " + str(coins) + " in coins back.")
54 elif bills == 500:
55     change = 500 - total
56     print("Your change is " + str(change) + " dollars")
57     coins = change % 1
58     coins = round(coins, 2)
59     ones = change % 10
60     ones = int(ones)
61     tens = change % 100
62     tens = tens / 10
63     tens = int(tens)
64     hundreds = change % 1000
65     hundreds = hundreds/100
66     hundreds = int(hundreds)
67     print("You will get " + str(hundreds) + " hundred dollar bills back.")
68     print("You will get " + str(tens) + " ten dollar bills back.")
69     print("You will get " + str(ones) + " one dollar bills back.")
70     print("You will get " + str(coins) + " in coins back.")
71 else:
72     print("You need to pay with a 10, 50, or 100 dollar bill")
```

