🕵️ PRACTICE PRODUCT - ONLINE STORE 🦜

🔨 BREAK IT DOWN

Let's say you have an idea for a product. You want to create an online store for selling your own line of Monty Python cosplay apparel! To tackle this efficiently, you would break it down into manageable components. Each major area includes smaller tasks or decisions you'll need to make.

Here is a possible breakdown of the main tasks you would need to make for your product: Business Planning, E-Commerce Platform & Technology, Website Design & UX, Product & Inventory Management, Logistics & Fulfillment, Marketing & SEO, Customer Support, and Analytics & Optimization.

Each of those tasks could be broken down further. If you took the E-Commerce Platform, you could break that down into smaller tasks such as: Architecture & Planning, Product Catalog System, User Authentication & Accounts, Shopping Cart, Checkout System, Order Management, Notifications System, Content Management, Search & Filtering, Analytics & Admin Dashboard, Security & Compliance, Testing & QA, and DevOps & Deployment

You would continue breaking down the problem into smaller more managable problems, and many could be completed by writing computer code. Take the Shopping Cart task for example. You could start coding a simple shopping cart with Python, where users can add items to the cart, remove items, view the contents of the cart, and see the total. You could store this collection of items in a Python list, output the list contents and totals, output a menu to add/remove items, and get input from the user to update the cart.

SHOPPING CART TASK

Focus on the task to create a program that stores a list of products in a shopping cart along with their prices. The user should have the ability to add items to the list, remove them, and see the total price of the cart.

TASK REQUIREMENTS

The user will be given a menu and have the ability to choose items from the menu. The options in the menu include the following:

1. Add a new item
2. Display the contents of the shopping cart
3. Remove an item
4. Compute the total
5. Quit

When the user chooses one of these options, the program should perform the appropriate action. Then the program should show them the menu again, and allow them to choose another option. It should continue running until the user choose the option to quit.

EXPECTED BEHAVIOR OF EACH OPTION:

ADD A NEW ITEM

The program asks the user for the name of the item and the price of the item.

The program stores these values in lists. You are storing both names and prices so you should use two lists, one for the names and one for the prices.

An example of the functionality of this item could look as follows:

Console Output

```
What item would you like to add? Bowler Hat
What is the price of 'Bowler Hat'? 65.99
'Bowler Hat' has been added to the cart.
```

After the user provides these values, the item and its price are each stored in a list.

DISPLAY THE CONTENTS OF THE SHOPPING CART

The program should display all of the items in the shopping cart, one per line. The price of each item should be displayed next to the item.

The program should also display the number associated with each item in the list, beginning with 1.

An example of the functionality of this item could look as follows:

Console Output

```
The contents of the shopping cart are:
1. Bowler Hat - $65.99
2. Police Whistle - $24.99
3. Bird Cage - $48.50
```

REMOVE AN ITEM

The user types in the number of the item they want to remove and the item is removed. Both the name and the price must be removed from their respective lists.

The following shows an example of this functionality assuming there are 5 items in the list:

Console Output

```
Which item would you like to remove? 3
Item removed.
```

Console Output

```
Which item would you like to remove? 13
Sorry, that is not a valid item number.
```

COMPUTE THE TOTAL

The program should iterate (loop) through each item in the list and add up the prices and then display the total amount to the user.

The following shows the expected result, assuming the total amount of the items in the shopping cart is $35.68

Console Output

```
The total price of the items in the shopping cart is $35.68
```

The following shows an example of the complete program:

Console Output

```
Welcome to the Shopping Cart Program!

Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 1
What item would you like to add? Cheese
What is the price of 'Cheese'? 3.49
'Cheese' has been added to the cart.

Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 1
What item would you like to add? Bananas
What is the price of 'Bananas'? 2.50
'Bananas' has been added to the cart.

Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
```

```
5. Quit
Please enter an action: 1
What item would you like to add? Spam
What is the price of 'Spam'? 4.00
'Spam' has been added to the cart.

Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 2
The contents of the shopping cart are:
1. Cheese - $3.49
2. Bananas - $2.50
3. Spam - $4.00

Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 3
Which item would you like to remove? 2
Item removed.
Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 2
The contents of the shopping cart are:
1. Cheese - $3.49
2. Spam - $4.00

Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 4
The total price of the items in the shopping cart is $7.49
```

```
Please select one of the following:
1. Add item
2. View cart
3. Remove item
4. Compute total
5. Quit
Please enter an action: 5
Thank you. Goodbye.
```

SHOWING CREATIVITY AND EXCEEDING REQUIREMENTS

In addition to the functionality listed above, add something of your own creativity to the assignment. You could consider adding better formatting (for example, aligning the prices), storing extra information, such as the quantity of the items in the list, or you can add anything else you think of to make the shopping cart better.

SUBMISSION

When the task is complete, upload your code to I-learn.