Hyperiondev

TASK

Capstone Project IV — Object-Oriented Programming

Visit our website

Introduction

WELCOME TO THE FINAL CAPSTONE!

Well done! This is the final Capstone Project for this bootcamp! This Capstone is a milestone in your learning so far. In this project, you will be using object-oriented programming to create a solution for a real-world problem. Remember, it is worth putting some extra time and effort into this project. It will eventually become part of your developer portfolio.

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to **www.hyperiondev.com/portal** to start a chat with a code reviewer. You can also schedule a call or get support via email.

Our expert code reviewers are happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

DEVELOPER PORTFOLIO

Developers who have the edge are those who find ways to apply their newfound skills from the get-go. A **developer portfolio** (a collection of online creations that you have made) allows you to demonstrate your skills rather than just telling people about them. It's a way of bringing your CV to life and introducing yourself to the world. As you learn more skills and put these into practice, each project that you complete will become more efficient and eye-catching.

Object-oriented programming is one of the most important programming paradigms today! Prospective employers will want evidence that a Software Engineer is comfortable using object-oriented programming. This application series offers you the means to an object-oriented program to add to your developer portfolio.

THE TASK AT HAND

Let us assume that you work for a Nike warehouse. Your role is that of a store master, meaning you are responsible for managing the warehouse, and more importantly, doing stock taking. To optimize your delivery time and for neater organization, you have decided to use your Python knowledge to get an overview of what each stock-taking session entailed.

Nike warehouses store the following information for each stock-taking list:

- Country
- Code
- Product
- Cost
- Quantity
- Value

Other store managers (in other regions) would like to be able to use your program to do the following:

- Search products by code.
- Determine the product with the lowest quantity and restock it.
- Determine the product with the highest quantity.
- Calculate the value of each item entry, based on the quantity and cost of the item. The value is calculated by multiplying the cost by the quantity for each item entered.

Before you begin

A key focus of this project will be ensuring that your code is correct, well-formatted and readable. In this regard, make sure that you do the following before submitting your work:

1. Make sure that you have identified and removed all syntax, runtime and logical errors from your code.

2. Make sure that your code is readable. To ensure this, add comments to your code, use descriptive variable names and make good use of whitespace and indentation. See **this style guide** to see how classes and methods should be named and how your program should be formatted.

3. Make sure that your code is as efficient as possible. How you choose to write code to create the solution to the specified problem is up to you. However, make sure that you write your code as efficiently as possible.

4. Make sure that all output that your program provides to the user is easy to read and understand. Labelling all data that you output (whether in text files or to the screen) is essential to make the data your program produces more user-friendly.

Compulsory Task

Follow these steps:

- Code a Python program that will read from the text file **inventory.txt** and perform the following on the data, to prepare for presentation to your managers:

    - Create a file named **inventory.py**, where a Shoe class should be defined.

    - In the Shoe class, create a function (**read_data()**) that will implement a try-except block for reading the following information from the file:
        - country,
        - code,
        - product,
        - cost, and
        - quantity.

    - Create at least 5 shoe objects and store these in a list. Add functionality to search products in the objects list by code.

    - Write code to determine the product with the lowest quantity, and restock it.

    - Write code to determine the product with the highest quantity and mark it up as being for sale.

    - You will have noticed that in the original data, there are only five columns. Write a function, **value_per_item()**, to calculate the value (or total worth) of each item entered. (Please keep the formula for value in mind; **value = cost * quantity**.) This function should then create a sixth column for each product, named value.

    - Represent the data from the objects list in a table format using Python's tabulate module, with the new value column added.

Rate us

Share your thoughts

Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.