



TASK

Capstone Project II

Visit our website

Introduction

WELCOME TO THE SECOND CAPSTONE TASK!

In this Capstone Project, you will be consolidating all the knowledge that you have gained in this level of the Bootcamp so far. You will extend the program that you have been working on in the previous Capstone Projects. This project should make it clear to a prospective employer that you can follow a development process to create an object-oriented program that is debugged, tested, refactored and documented program that meets a client's specification!



Get in touch
Connect for support

Remember that with our courses, you're not alone! You can contact your mentor 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 your mentor. You can also schedule a call or get support via email.

Your mentor is 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

By the end of this project, you will be adding the program that you have been developing throughout this level to your developer portfolio. This project should make it clear to a prospective employer that you can follow a development process to create an object-oriented program that is debugged, tested, refactored and documented and meets a client's specification! Put extra time and effort into this project to make sure that it really showcases the skills you have acquired!

THE TASK AT HAND

You are asked to create a food delivery system for a company called "Food Quick". Food Quick is the company that receives the orders and distributes them to a driver based on their current load and their location. They want you to create a program that can help them keep track of the orders and distribute accordingly.

Food Quick stores the following information for each customer:

- Order number
- Customer name
- Contact number of the customer
- Address of the customer
- Location (city) of the customer
- Email address of the customer
- Name of the restaurant
- Location of the restaurant
- Contact number of the restaurant
- How many of each meal is being ordered
- The list of meals being ordered and their prices
- Any special preparation instructions given by the customer
- The total amount to be paid

The information about the drivers is in the text file **drivers.txt** in the following format:

John Krill, Cape Town, 4

This shows that the driver's name is John Krill who is in Durban and he currently has a load of 4 deliveries.

Food Quick would like you to be able to create an invoice for a customer after the above information has been inputted into the program. The invoice should be a text file with the following format:

Order number 1234

Customer: Jill Jack
Email: jilljack@yahoo.com
Phone number: 123 456 7890
Location: Cape Town

You have ordered the following from Aesop's Pizza in Cape Town:

1 x Pepperoni pizza (R78.00)
2 x Hawaiian pizza (R82.00)

Special instructions: Extra tomato base on the Pepperoni pizza

Total: R242.00

John Krill is nearest to the restaurant and so he will be delivering your order to you at:

12 Cherry Road
Plumstead

If you need to contact the restaurant, their number is 098 765 4321.

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 1

Follow these steps:

- Design your program to meet the specifications given by the client. Extend the program that you have written in the previous Capstone Project so that this program also:
 - Outputs a text file that shows a list of the customer's names and order numbers in alphabetical order
 - Outputs a text file that shows a list of the customer's names grouped by location.
 - Updates the original **drivers.txt** file to add to their load if they have been matched with a customer. E.g. if a line reads **John Krill, Cape Town, 4** and he gets paired with a customer to deliver, the line should be changed to say **John Krill, Cape Town, 5**.

Compulsory Task 2

Follow these steps:

- Enhance your program by doing the following:
 - Incorporate exception handling to your code. Your code should include at least two try-catch blocks (*Go back to the Towards Defensive Programming II task if you're not sure where to start*).
 - Make sure all your code is properly debugged. Make sure that the runtime and logical errors are also identified and corrected.
 - Fix the indentation and formatting of the code so that it adheres to the guidelines provided [here](#).
 - Make sure that all the names of variables, classes, methods, etc. adhere to the guidelines provided [here](#).
 - Refactor the code to improve the quality and readability of the code in other ways highlighted in this project.



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.

