Hyperiondev

TASK

Capstone Project - Object-Oriented Programming

Visit our website

Introduction

WELCOME TO THE OBJECT-ORIENTED PROGRAMMING CAPSTONE PROJECT!

Welcome to the first Capstone Project for this level! 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. You will be extending this Capstone Project in later Capstone projects. Remember, it is worth putting some extra time and effort into this project. It will eventually become part of your developer portfolio.

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

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.

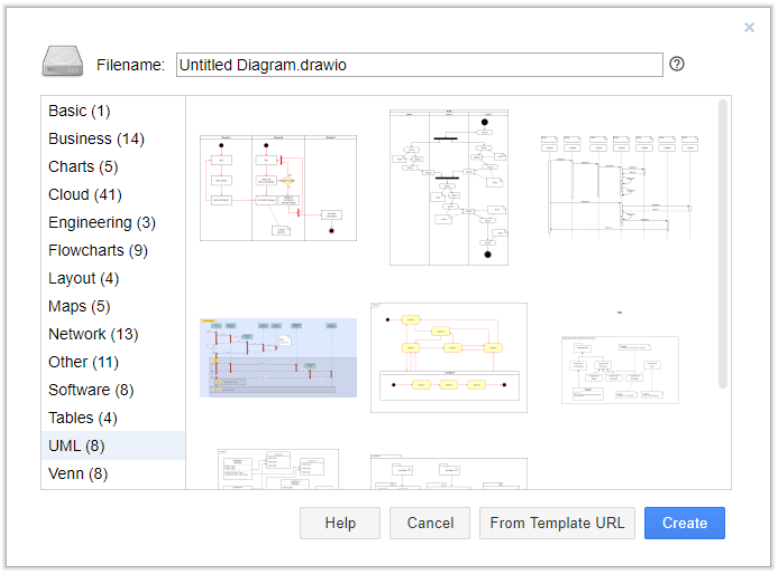
4. Make sure that all output that your program provides to the user is easy to read and understand. Labelling all data that you output (whether in text files or to the screen) is essential to make the data your program produces more user-friendly.

Take note:

Various tools can be used to create dependency diagrams. One such tool is **Draw.io**, a free and open source cross-platform graph drawing software developed in HTML5 and JavaScript. Its interface can be used to create diagrams such as flowcharts, wireframes, UML diagrams, organisational charts, and network diagrams

To use draw.io to create dependency diagrams:

- Open **https://www.draw.io/**
- When a blank, untitled diagram is opened, click on the "+ More Shapes" button at the bottom left-hand corner of the screen.
- From the dialogue that appears, select "UML" and then press "Create".



- To add a class diagram, select the class diagram shape from the toolbox as shown in the image below:

- While Draw.io provides an intuitive interface for designing and creating high-quality diagrams, they also provide interactive tutorials and step-by-step guides on their **Tutorials Page**.

Compulsory Task 1

Follow these steps:

- In a file called **planning.txt**, design your program to meet the specifications given by Quick Food. Write out the pseudocode of how you plan for your program to run. Include the creation of classes and the methods you plan to have in those classes. You should also show the relationships between classes. You will not be marked on this, it is just to help you create a clear plan before creating your code.
- Next, use your pseudocode to create a UML diagram and save it as a PDF in this task's folder. You should now have a good idea of how your program is going to flow.
- **Reminder**: this compulsory task should be **console/terminal-based**.

Compulsory Task 2

Follow these steps:

- Code a Java program that will meet part of Quick Food's specifications. You will build upon this program in later Capstone projects. For this program, you should:
    - Create a class that will be used to create a Customer object.
    - Create a class that will be used to create a Restaurant object.
    - Write a program that will allow a user to:
        - Capture the details that are used to create a new Customer object.
        - Capture the details that are used to create a new Restaurant object.
        - Read the **drivers.txt** file and find the driver in the correct area with the smallest load. *Note, you just need to check if the driver and the restaurant are in the same area.*

- If the customer lives in a location that does not match a driver's location, the invoice only needs to say: **"Sorry! Our drivers are too far away from you to be able to deliver to your location."**

- Write up the invoice to a new text file for the customer called **invoice.txt**.

Reminder: this compulsory task should be **console/terminal-based**.

Rate us

Share your thoughts

HyperionDev 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.