

CLASSES & OBJECTS

LAB

Tech Stack

Python

Notes

Each set of tasks represents a new “program” which consists of a set of classes that will be instantiated into objects that interact with each other.

What is a Class?

- A class is a template for an object. It defines what the object is made up of (class properties) and what actions the object can perform (member methods)

What is an Object?

- An object is an instance of a class. Each instance of a class (object) contains the same class properties and methods. This means each instance of a class (object) has the same makeup. What differs between the objects is what values are set equal to the member variables.
- To put this in perspective of what you are more familiar with, every time you run an application on your computer it is running an instance of that application. For example, you can have multiple Microsoft Word documents open at once. Each of these Microsoft Word documents may contain different text and fonts, but all of them have the same functionality.

Setup Steps

- ☐ Create a new Python project in Visual Studio Code
- ☐ Create a file called **main.py** - this is where you will test all of your classes
- ☐ Create the following files for each set of tasks:
 - ☐ Alarm Clock: **alarm_clock.py**
 - ☐ Customer Shopping Cart: **customer.py, shopping_cart.py, product.py**
 - ☐ Cell Phone (Bonus): **cell_phone.py**

Alarm Clock Tasks

- As a developer, I want to use Python's proper snake_case for variable names.
- As a developer, I want to create an AlarmClock class (alarm_clock.py).
- As a developer, I want the AlarmClock class to have class variables to keep track of the AlarmClock's current time, whether the alarm is on or off, and the time the alarm is set to.
 - NOTE: *You can use arbitrary strings to represent the time, it does not need to be accurate or change over time*
- As a developer, I want the AlarmClock class to have a method to set (or change) the current time and print to the console the current time.
- As a developer, I want the AlarmClock class to have a method to toggle the alarm on or off.
- As a developer, I want the AlarmClock class to have a method to set the current alarm time and print to the console the current alarm time.
- As a developer, I want to import the AlarmClock class into main.py so I can instantiate it as a new AlarmClock object and call methods on it.
 - Print the clock's time to the terminal
 - Call the alarm clock's change time method to change the time
 - Toggle the alarm clock's on off switch

Customer Shopping Cart Tasks

- As a developer, I want to use Python's proper snake_case for variable names.
- As a developer, I want to create a Customer, ShoppingCart, and Product class.
- As a developer, I want the Product class to have class properties to keep track of the Product's name, price, and category.
- As a developer, I want the Product class's initializer to take in the initial values for the name, price, and category via parameters
- As a developer, I want the ShoppingCart class to have class properties to keep track of the ShoppingCart's products (list).
- As a developer, I want the ShoppingCart class to have a method to calculate and return the current total of all products in the cart.
- As a developer, I want the ShoppingCart class to have a method to add a new product to the cart. (Appending to the products list)
- As a developer, I want the ShoppingCart class to have a method to empty all products from the shopping cart.
- As a developer, I want the Customer class to have class instance variables to keep track of the Customer's name and shopping cart object. Set the shopping cart instance variable equal to a new ShoppingCart object in the initializer.
 - HINT: You will have to import the ShoppingCart class into the customer.py file
- As a developer, I want the Customer class's initializer to take in the initial value for the customer's name via a parameter.
- As a developer, I want the Customer class to have a method to add a new product to the customer's shopping cart (within this method you will call the shopping cart's add product method)

- As a developer, I want the Customer class to have a method to view all products in the customer's shopping cart. (Loop over the shopping cart's products list and print each product to the terminal)
- As a developer, I want to import the Customer and Product classes into main.py so I can instantiate a customer object as well as three Product objects and interact with the object's methods:
 - Print the customer's name to the terminal
 - Call the customer's add product to shopping cart method three times and add the three products objects you created
 - Call the customer's view products method
 - Call the customer's shopping cart's get cart total method. Capture the total the method returns in a variable and print to the terminal
 - Call the customer's shopping cart's empty cart method
 - Check if all products have been removed from the shopping cart

BONUS: Cell Phone Tasks

- As a developer, I want to use Python's proper snake_case for variable names.
- As a developer, I want to create a CellPhone class.
- As a developer, I want the CellPhone class to have class instance variables to keep track of the CellPhone's model, phone number, contacts (dictionary), messages (list) and whether the cell phone is on vibrate mode or not.
- As a developer, I want the CellPhone's model type to be passed into the classes' initializer via a parameter.
- As a developer, I want the CellPhone class to have a method to receive a text message that prints the message and adds it to the messages list
- As a developer, I want the CellPhone class to have a method to toggle whether the phone is in vibrate mode or not.

- As a developer, I want the CellPhone class to have a method to create and send (print) a new text message to a contact.
- As a developer, I want to import the CellPhone class into main.py so I can instantiate it as a new CellPhone object and interact with it:
 - Print the cell phone's contacts to the terminal
 - Send two new messages to the phone through the receive text message method
 - Print the cell phone's messages to the terminal
 - Call the create text message method to create a new message
 - Toggle the cell phones ringer to vibrate
 - Print the cell phone's current ringer/vibrate setting to the terminal