# Lab 12-2 – Barista OOP

## Goals

- Practice with object oriented programming
- Practice creating classes with methods and attributes
- Practice creating class variables and instance variables
- Using instance variables and their associated methods

## Setup

- Create a new Python project using the following naming convention
  **`ITP115_L12_2_`*`Lastname_Firstname`***
  (replace *Lastname* with your last/family name and *Firstname* with your first name)
- Add the **`CoffeeShopProgram.py`** file into your project
- Create a Python file named **`Coffee`**
- Create a Python file named **`Barista`**
- Your new files must begin with comments in the following format *(replace the name and email with your actual information):*

```
# Name, USC email
# ITP 115, Spring 2020
# Lab 12-2
```

## Requirements

In this lab, you will create two classes to model a coffee shop interaction with a barista. You will be defining two classes, **`Coffee`** and **`Barista`**. The Barista class will contain an attribute which is a list of Coffee objects.

Provided for you *(do not modify)*

- **`main`** is completed for you already in a file named **`CoffeeShopProgram.py`**. Be sure to place the file in the same directory as your other two files. Since it is already defined, your class methods need to be correctly defined and the exact naming conventions of the class methods must be used in order for the correct output to occur. Here is a brief description of what main does:
  - Creates a **`Barista`** object.
  - Takes coffee orders from the user as long as the barista is still accepting orders. This is accomplished by using the **`Barista`** class method **`isAcceptingOrders`**.
  - The barista "makes" and serves all of the drinks after accepting 5 orders using the **`Barista`** class method **`makeDrinks`**.
  - Ask the user if they would like to continue.

What you need to write

- **`Coffee`** Class, define this class in a file called **`Coffee.py`**
  - Class variable:
    - Create a class variable called **`STATUSES`** which is a **`list`** containing the strings "ordered" and "completed". You will use this class variable in the string method as described below.
  - Instance attributes/variables (to be assigned in **`__init__`**).
    - **`size`**: the size of the drink (small, medium, large)
    - **`desc`**: description of the drink (a string)
    - **`customer`**: the name of the customer who ordered the drink (a string)
    - **`statusIndex`**: 0 if the drink has not been made, 1 if the drink is completed. When new coffee objects are created, the **`statusIndex`** will be set to 0 initially
  - Constructor Method:
    - **`__init__`**
      - Inputs (3): size of the drink, a description of the drink, and the name of the customer who ordered the drink
      - Return: none
      - Set the size, description, and customer attributes of the coffee object. Set the **`statusIndex`** attribute to 0 (representing an uncompleted drink)
  - Instance Methods:
    - **`setCompleted`**
      - Input: none
      - Return: none
      - Set the **`statusIndex`** attribute to 1, indicating that the drink is completed
    - **`__str__`**
      - Input: none
      - Return: a string containing information about the coffee object
      - The message should be a nicely formatted string with information about the coffee object's size, description, the customer who ordered it, and status. Be sure you use the class variable, **`STATUSES`**, in the message (use in combination with **`statusIndex`** to get the correct string version of the status).

- **`Barista`** Class. Define this class in a new file called **`Barista.py`**
  - Class variable:
    - Create a class variable called **`MAX_ORDERS`** and set its value to 5. This represents the maximum number of drinks the barista can keep track of at a time.
  - Instance attributes/variables (to be assigned in **`__init__`**).
    - **`name`**: a string representing the barista's name
    - **`orders`**: a **`list`** of **`Coffee`** objects representing the drinks the barista has to make.
  - Constructor Method:
    - **`__init__`**
      - Input: (1) the name of the barista
      - Return: none
      - Set the name of the barista. Set the orders to an empty list.
  - Instance Methods:
    - **`takeOrder`**
      - Input: none
      - Return: none
      - Take a coffee order by asking the user to input their name, the size of the drink, and the type of the drink that they are ordering. Use these three inputs to create a new **`Coffee`** object and add it to the list of orders. Print out the created drink.
    - **`isAcceptingOrders`**
      - Input: none
      - Return: **`Boolean`**
      - Return **`True`** if the barista can still take more orders (i.e. the list of **`Coffee`** objects is less than **`MAX_ORDERS`**). Return **`False`** otherwise.
    - **`makeDrinks`**
      - Input: none
      - Return: none
      - Go through the list of orders and "make" all of the drinks by using the **`Coffee`** class's **`setCompleted`** method. Print out each completed order.
      - After making all of the drinks, reset the barista's list of orders back to an empty list.
    - **`__str__`**
      - Input: none
      - Return: a string containing a message with the barista's name

## Sample Output

```
Please enter your name: Trina

~ Welcome to the Coffee Shop! ~

Hello, my name is Trina, and I am your barista.
What drink do you want? flat white
What size would you like? small
What is your name? Dhivya
small flat white for Dhivya (ordered)

Hello, my name is Trina, and I am your barista.
What drink do you want? chai latte
What size would you like? medium
What is your name? Lucia
medium chai latte for Lucia (ordered)

Hello, my name is Trina, and I am your barista.
What drink do you want? macchiato
What size would you like? medium
What is your name? Mehr
medium macchiato for Mehr (ordered)

Hello, my name is Trina, and I am your barista.
What drink do you want? latte
What size would you like? large
What is your name? Rachel
large latte for Rachel (ordered)

Hello, my name is Trina, and I am your barista.
What drink do you want? coffee
What size would you like? small
What is your name? Andrew
small coffee for Andrew (ordered)
```

```
Here are the completed orders:
     small flat white for Dhivya (completed)
     medium chai latte for Lucia (completed)
     medium macchiato for Mehr (completed)
     large latte for Rachel (completed)
     small coffee for Andrew (completed)
Would you like to continue (y/n)? n
Goodbye, please come again!
```

## Deliverables and Submission Instructions

- Create a zip file containing your Python code. This cannot be done within PyCharm. Find the folder on your computer and compress it.
  a. Windows*:*
     1. Using File Explorer, select your lab folder
     2. Right click
     3. Send to ->
     4. Compressed (zipped) folder
  b. Mac OSX:
     1. Using Finder, select your lab folder
     2. Right click
     3. Compress "*FileName*"
- Upload the zip file to your Blackboard section:
  1. On Blackboard, click on the Labs item in the course menu on the left.
  2. Click on the specific item for this assignment (starts with L and a number).
  3. Click on the Browse My Computer button and select your zip file.
  4. Click the Submit button.