ACIT 2515 – Object Oriented Programming - Lab 1

Instructor	Mike Mulder (mmulder10@bcit.ca)
Total Marks	20
Due Dates	Monday, May 16th at midnight **No Late Submissions Will Be Accepted**

Goals

- Setup and use the development tools for this course.
- Create your first Python class for the course.

Prerequisites

Please make sure you have the following software installed on your computer:

- Python 3 (3.9 or later) installed
- Pycharm or Visual Studio Code installed

Make sure you sign-up for the class Discord channel as well. Discord is the preferred communication tool for labs and will be required for any lab demos.

Your First Python Class (20 Marks)

Instructions

1. Create a new project in PyCharm called Lab1.
2. Create two python files:
    a. instructor.py
    b. main.py
3. Create the Instructor class in instructor.py as per the specifications below.
4. Create the main program in main.py as per the specifications below.
5. Record the output of your main program (i.e., in PyCharm) via a screenshot.
6. Create a zip file called Lab1B.zip containing the following content:
    a. instructor.py – Instructor class
    b. main.py – Main program
    c. output.png – Screenshot of the output of your main program

Submission

Upload your zip file (lab1.zip) to the Lab1 dropbox (Activities -> Assignments -> Lab1) in D2L before midnight on Monday, May 16th.

Specifications

Requirement	Marks
Create a class named Instructor.	1
Include a class comment block and a comment block for each method using DocString (see below – Class Documentation Best Practices).	2
Follow basic Python best practices for naming (see below – Class Naming Best Practices).	2
Constructor that takes in and sets the following private attributes: • First Name • Last Name • Program (i.e., CIT) • Employee Number (i.e., A01072133) Name all instance variables with a leading underscore (i.e., _program).	2
Method that adds a course to the instructor, where the course is the ID. For example: ACIT2515 or COMP1510. **Note**: *Make sure you cannot add a course with the same ID more than once.*	2
Method that removes a course from the instructor based on the course ID (see above).	2
Method that checks if the instructor is teaching a course based on the course ID. Returns a boolean True or False.	1
Method that returns the full name of the instructor – i.e., returns a single string containing the First Name and Last Name separated by a space. For example, an instructor with first name of "Larry" and last name of "Jones" would return "Larry Jones" from this method.	1
Method that returns a string with the instructor details as follows: *Bill Smith is an instructor in the CIT program with employee number A01072133 teaching the following courses: ACIT2515, ACIT2520* OR *Bill Smith is an instructor in the CIT program with employee number A01072133 teaching no courses* Make sure to use the method you created to get the instructor full name when constructing the above string.	2

Main program that tests the Instructor class as follows:
• Creates an instance of the Instructor class for "Mike Mulder":
  ○ Add the following two classes: ACIT2515, COMP1516
  ○ Remove ACIT2515.
	4

○ Check if the instructor is assigned to ACIT2515. If he is not, print "I should be enrolled in ACIT2515!". ○ Print the instructor details. • Creates an instance of the Instructor class for "Susan Lee": ○ Add 5 classes to the instructor, including MATH1511. ○ Remove all but 2 classes (note: keep MATH1511). ○ Try to add MATH1511 a second time. ○ Print her instructor details. • Creates an instance of the Instructor class for another of your instructors: ○ Do NOT add any classes for this instructor. ○ Print his/her instructor details.	
Screenshot of the output of your program (as above).	1
Total	**20**

Documentation Best Practices

Use the following documentation practices below for this lab.

Class Documentation	Add a comment describing what the class represents.
	
	Use DocString as per the following example:
	
	`class Point:`
	`  """Represents a point in 2D geometric coordinates"""`
	
	`  def __init__(self, x=0, x=y):`
	`  ...`

Method Documentation	Add a comment describing what the method does.
	
	`def __init__(self, x=0, x=y):`
	`  """Initialize the position of a new point. The x and y`
	`  Coordinates can be specified. If they are not, the`
	`  point defaults to the origin. """`
	
	`def move(self, x, y):`
	`  """Move the point to a new position in 2D space. """`
	`  self.x = x`
	`  self.y = y`

Docstring Reference: https://www.python.org/dev/peps/pep-0257/

Naming Best Practices

Use the following naming practices for this lab.

Class Name	CapitalizedWords (aka CamelCase)
Instance Variables	lower_case_with_underscores **Note:** Use _lower_case_with_underscores for internal (i.e., private) instance variables.
Methods	lower_case_with_underscores **Note:** Use _lower_case_with_underscores for internal (i.e., private) methods.

Reference Style Guide: https://www.python.org/dev/peps/pep-0008/