

Lab 8 - Factory

Rahul Kukreja & Jeffrey Yim

jyim3@bcit.ca

Welcome!

Welcome to the 8th COMP 3522 lab.

In today's lab, you will:

1. Implement the Factory Pattern
2. Re-visit some old code you wrote and improve it with all the new knowledge you have gained.

Grading

This lab and all future labs will be marked out of 10

For full marks this week, you must:

1. (6 points) Implement the factory pattern correctly
2. (4 point) Improve the quality of your old code.
3. **Optional:** Draw a UML Class Diagram representing the system.

Requirements

Clone the project from: <https://classroom.github.com/a/zB-BCqKI>

For this lab we are going to re-visit old code that we wrote all the way back in Week 3.

Yep, the library / catalogue system.



This is very **IMPORTANT**. Read through the whole document before starting with the first task. **If you do not you might do unnecessary extra work in Task 1.**

If it makes things easier for you, you may work on both parts simultaneously or tackle Part 2 before Part 1. Every one's code is different which makes Part 1 different for everyone.

Part 1: Make-Over

Open up PyCharm and take a look at the Library Lab. It can be hard reading old code (hopefully you commented it well!). Duplicate the modules and move them over to the Lab 8 folder

I want you to improve the code you have written. The scope and ways you could improve will vary for each student. Some ways you could improve the code are:

- Using better data types
- Adding in properties and removing unnecessary getters and setters
- Comprehensions
- Using built in methods
- Refactor existing methods and classes.
- `**kwargs` and `*args`

Every time you have finished improving a part of your code commit it using Git and add in the improvement as the commit message. I will be going through your git commits to see what improvements have been made. Re-visit your feedback for some tips on where to get started.

NOTE: Avoid improving the `LibraryItemGenerator`. We are going to re-write the whole class.

Part 2: Item Factories

For the second task, you need to implement the Factory Pattern. You should have your `Item` Product hierarchy already set up. If this isn't an ABC, make it one now.

Each factory should be responsible for getting the right input from the user and generating the item with that input. Proper use of `**kwargs` in the `Item` hierarchy can help you re-use code in the factory classes.

If you didn't have a menu, then add one now. If you already had a menu then you might need to re-structure the Add Item section.

- Ensure you push your work to github classroom. I'd like to see sensible git commits comments, and commits must take place at logical points in development.