**Name**: Damian Zukowski

**Pair:** Me, myself and I

**Amount of completed tasks:** 7

**Which tasks were left undone or incomplete:** 5, 6 and 7

Self-assessment:

This exercise was ok/etc. for me because… I learned classes and I believe I understand them well enough. All I need is more practice.

Doing this exercise, I learned… much more about private attributes and methods and how to access and use them. The @property “getter” option or the “.setter” are really helpful.

I am still wondering… about the more complex tasks where more objects are involved and the complex methods they can contain. That’s the hard part that requires a lot of thinking on my part.

I understood/did not understand that… ; I did/did not know that… ; I did/did not manage to do…

I did not manage to do 3 tasks as I simply ran out of time ☹ But I will do them in my own time later on, because they are more complex and would be great for learning.

## Answers to other than coding tasks here:

1. a) Encapsulation is the process of hiding the internal state of an object and requiring all interactions to be performed through an object’s methods. This approach:

Provides better control over data. Prevents accidental modification of data. Promotes modular programming.

Python achieves encapsulation through **public**, **protected** and **private** attributes.

b) A client is a program that’s written in Python and makes use of functions in a module.

c) Attributes in Python are characteristics of an object – such as variables or methods – that store data about its properties; used to describe or quantify an object or entity. They can be class attributes (belongs to a class) or instance attributes (belongs to a specific instance/object).

d) An instance of a class in Python is called an object. The instance is created based on a class as a template and then it has a life of its own as an object.