CSF202 – Lab Book

Tom Blanchard with thanks to Mike Edwards

This Lab Book contains the lab tasks for CSF202 which are designed to be completed during the lab class each week.

Each week you will be given a new lab class sheet with new tasks. You should aim to get all tasks completed and marked off by staff in the lab class. If you do not get the tasks finished in the lab class then you should have them completed for the next lab session.

Full marks will be awarded if the tasks are marked off by staff during the session they are handed out; or if they are marked off by staff during the following session. Half marks will be awarded if the tasks are signed off later than this.

The lab work as a whole is worth 10% of the module.

CSF202 – Lab Class 1 – Wednesday 23/09/2020

□ Task 1.1

This task involves writing two Java classes. The purpose of this task is to recap and get you familiar with Java and objects once again.

We want to model two entities: A person and a lecturer. There is an inheritance relationship between the two: decide which is the superclass and which is the subclass.

A Person has the following attributes:

- Name (String), this is mandatory.

A lecturer has the following **additional** attributes:

- Staff ID (Integer), this is mandatory.
- Office location (String), For example “Computational Foundry 413”. This is optional.

Create Java classes to model these two entities. You should only be able to create a person if you provide their name. Similarly, you should only be able to create a lecturer if you provide both their name and their staff ID.

Your classes should have setters and getters for all fields (unless you can think of specific reasons for not allowing them to be set after creation).

Place a print statement in each constructor so that you can see the constructors being executed and in which order they are executed in order.

Test your classes in your main method. You should create at least two objects: a person and a lecturer and play around with the setters and getters (and print statements) to make sure they work as expected.

□ Challenge Task 1.2

The office location in Task 1.1 is a bit simplistic. Really speaking this should be an additional class with at least the following two instance variables:

- Building (e.g., “Tom’s Flat”)
- Room (E.g., “The Living Room”)

Create such a class and then alter the lecturer class so that it uses your new class. Remember that instances of this class can be passed as parameters in constructors and methods.

□ Challenge Task 1.3

In which ever class you chose to be the subclass try overriding one of the superclass's methods. What happens what you call this method through a superclass reference? For example if you chose Person as your superclass and Lecturer as your subclass:

- Person p = new Lecturer();
- p.overriddenMethod();

Does this behave the way you expected?