University of HUDDERSFIELD

CFS2160: Programming Stream

Tutorial/Practical 7

Welcome to Java

Introduction

We are now moving on from Python to Java.

As usual, activities marked ⇶ should be included in your logbook for assessment. All code *from now on* should be submitted as *listings*, using the layouts previously given. All listings should be accompanied by screenshots showing the programs running.

Purpose

By the end of the week you should have a complete up-to-date Python logbook that will require little, if any, attention between now and submission time.

You should also have started to get to grips with the basics of Java.

As usual:

If in doubt - Ask!

Activities

1. Make sure your Python work is complete. Ask your friendly tutor if you need any advice or guidance.

2. Start IntelliJ (you may also want to download it for your own use). Find the "Hello, World" Java program from the lecture slides, and make sure you can enter and run it. The program from the lecture actually implements a *class* that has a single *constructor* that uses a *method* that in turn prints the message. Make sure you understand what those italicised words mean.

3. ⇶
   In the GitHub repo for the Java part of the module[1] you will find a simple Java class called `Employee`. Examine the class, and record the names and types of all the instance variables (attributes). Now look at the methods, and for each record its name, what it

[1] https://github.com/TonyJenkins/cfs2160-2018-java-public.git

does, and what type it returns.
Enter your notes in your logbook.

4. ⇶
If you didn't do the second task above because there were no arrows next to it, go back and do it now.
The `Employee` class has a blank `main` method at the end. Copy it into IntelliJ and Implement a short program here that does the following. Print the employee out after each operation, and include the code and output in your logbook.
Note that you will probably need to change the package at the top of file to match the structure you have chosen for your own source code.

   - Creates a `Employee` object with an id of "123" and a name of "Jane Smith".
   - Sets Jane's salary to £24000.
   - Sets the name of Jane's manager to "Barbara Forbes".
   - Gives Jane a 5% salary raise.
   - Displays *just* the new salary of the employee.
   - Sets Jane's job title to "Senior Account Executive".
   - Displays whether or not Jane is allowed a parking permit.

Hint: You saw everything you need to do this in the lecture today, except the class used there was different.

The output from the final task should resemble the below.

```
Creating an Employee ...
Employee{id=123, name='Jane Smith', jobTitle='null', manager='null', salary=0}

Setting Salary ...
Employee{id=123, name='Jane Smith', jobTitle='null', manager='null', salary=24000}

Setting Manager ...
Employee{id=123, name='Jane Smith', jobTitle='null', manager='Barbara Forbes',
salary=24000}

Applying 5% Salary Raise ...
Employee{id=123, name='Jane Smith', jobTitle='null', manager='Barbara Forbes',
salary=25200}

Jane's Salary is £25200.

Setting Job Title ...
Employee{id=123, name='Jane Smith', jobTitle='Senior Account Executive',
manager='Barbara Forbes', salary=25200}

Finding Car Parking Status ...
Employee is permitted a car parking space.
```