

121COM: Introduction to Computing

Academic Year 2015/16

LabSheet 4

For use in labs the week beginning Mon 19th October 2015

Author: Dr Matthew England (Coventry University)

Most of the questions on this lab sheet require you to write a function. When you do, make sure you also write some documentation (docstring, comments)!



Lab Exercises 1 - Not too taxing

These questions concern the UK tax system. All rules / figures were correct at the time of writing (Sep '15).

Write the functions for these questions in a single script called `Week4Lab1.py`. On Moodle there is a file `Week1Lab4Tests.py` which will test your functions from Q1–3. Save the test script into the same directory as your answer script and run it when you are ready to test. Note that tests expect all input and output of your functions to be integers (which makes sense as in the UK all taxes are rounded to the nearest pound).

- Q1 UK tax payers receive a **Personal Allowance**: income on which they do not pay income tax. For most people this is £10,600, but the allowance reduces by £1 for every £2 that your income is above £100,000. Write a function `personalAllowance` which takes an annual salary and returns a personal allowance.
- Q2 UK Income Tax is paid on your taxable salary (salary after personal allowance). You pay 20% on the first £31,785 of taxable salary; 40% on any taxable salary above this and below £150,000; and 45% on anything more. Write a function `incomeTax` which takes a users annual salary as input and returns their income tax bill. It should call your `personalAllowance` function.
- Q3 UK National Insurance (a mandatory payment for state benefits such as the NHS) is paid based on your weekly salary. You pay 0% on the first £155; 12% on what you earn above this and below 815; and 2% on anything more. Write a function `nationalInsurance` which takes a weekly salary as input and calculates weekly national insurance contributions as output.



Extensions: If you want to get a real idea of your monthly pay cheque you'll need to do the following two questions also!

- Q4 Most people in the UK are paid monthly. Write a function `monthlyPay` which takes a users annual salary and gives two outputs: how much money they receive a month after tax; and their total monthly deductions. The tax authorities work with the average of 4.33 weeks in a month.

Q5 Those with a student loan from the UK government also have to make repayments monthly together with their tax. They make a repayment of 9% of their monthly salary above £1,444. Write a fourth function `studentLoan` to calculate monthly student loan repayments.

Copy your function definition from Q3 and call the copy `monthlyPayExt`. Extend it so that it now takes two inputs. The second should be a Boolean indicating whether the individual has a student loan.



Lab Exercises 2 - More fun with functions

Before starting to write the functions for Q6 and Q7 first: (a) sketch a flowchart for your functions; (b) write some simple tests that your function should pass. When writing the tests, try to think of the different branches in the flowchart that could be triggered.

Q6 Write a program that asks a user for an integer between 1 and 9999. The program should print the equivalent to the input in English. For example, if the number was 1234 it should print `one thousand two hundred and thirty four`. Do not use the Standard Library for this problem.

Q7 A Caesar Cipher is a way of sending secret messages. It uses an integer between 1 and 26 as a key. Each letter in a message is moved forward by that many letters (with letters at the end of the alphabet going to the start). For example, if the word is `yellow` and the key 3 then the message becomes `bhoorz`.

Write a script which when run first asks the user for an integer between 1 and 26 (the key). It should allow the user to either `encode` a message or `decode` a message. It should allow the user to try unlimited messages until they type `exit`.



Extension: Extend your cipher so instead of just working on 26 letter alphabet it also encodes with capitals, numbers, spaces etc.

Q8 When Python encounters a variable it looks for the value assigned in various **namespaces**: the local function, any outer functions (if nested), the main script, the built-in commands. Write your own simple examples which prove to you the order in which Python looks through these.



Extended Task - Automata

This extended task concerns Finite State Automata (see Week 4 Slides C). There is a Java executable available to download from Moodle called JFLAP. When run choose **Finite Automaton** from the menu. This starts a simple program where you can create diagrams of Finite State Automata like those described in the slides; and then see how they behave under different input strings. The program is fairly self explanatory, but if you need help there is a tutorial online here: <http://www.jflap.org/tutorial/>

Create automata that work on binary strings and accept the following languages:

- Strings with 11 as a substring.
- Strings with 11 or 00 as a substring.
- Strings that begin with 1 or end with 1.
- Strings that begin with 1 and end with 1.

Consider a FSA that would control traffic lights at a crossroads. There are two sets of lights (those for horizontal traffic and those for vertical). Each set of lights can be either green, amber or red. How many states will you need? The simplest model will cycle through the states stopping for given at each for periods of time - so only one possible input (the passing of that time). However, you might consider how the model would be extended to respond to different traffic conditions; or the need for pedestrians to cross.

Another real-world machine that runs via FSA is a lift. Can you design an FSA for the lift in the ECB?