PAL Introduction *University of Hertfordshire – Week 1 – October 2015 –*

1 Aim & Objectives

- Introduce PAL
- Ice-Breaker
- Future PAL sessions
- Programming first steps

2 Introduction

2.1 What is PAL?

- An opportunity for experienced students to re-invest into first year students
- Collaborative learning, i.e. you get as much out as you put in

2.2 What does PAL do for me?

- Adaptive pace to suit your abilities
- Student orientated sessions from the perspective of students
- Answer questions about programming (not coursework)

2.3 What to expect in future sessions...

Python! Lots and lots of Python! It probably won't be too surprising that in your programming sessions you will mostly be programming. The difference between these sessions and your other programming sessions will be the pace offered, where in your normal sessions you will be expected to learn at the same speed as the entire course. These sessions are amied to be tailored to your needs, we'll re-cover difficult subjects and move onto more advanced programming if you find the content too easy.

3 Exercise - Getting Started

In this exercise we aim to simply get you setup with idle, the standard IDE since version 1.5.2b1. We will be using Python version v3.X, the newer the better. (At the time of writing there isn't a stable v4.X.)

To do this, you'll have to locate the idle environment. In whichever OS (Operating System) you have chosen to use, simply search for 'idle'. If this is successful and you're presented with a console (command line, i.e. window with monospace text inside). Please check that the version is greater than v3.0.

If the computer you're working on doesn't have this, we can install it simply at '*http://www.python.org/downloads/*'. If you happen to be using a Unix based operating system, please refer to your trusty package manager (look for '`idle3`').

Now select '*File/NewFile*'. This is where you'll be typing all of your code. To run some code, select '*Run/RunModule*'. For now this won't do anything interesting to us.

4 Exercise - Hello World

Please navigate to the '*http://homepages.herts.ac.uk/~db12aba/*' site, where we'll be uploading all of the code and sheets for these sessions. Here, you should see a section named 'Resources' with 'Week 1' underneath. Now select the 'helloworld.html' link. From now on we'll just format this as '*Resources/Week1/helloworld*' for simplicity.

Here we have it, the classic '`"Hello World!"`' example that most programmers begin with. It's history stretches back before the C language (not C++!) where Brian Kernighan (one of the chaps that wrote the C book in 1972) is said to have written it as part of the documentation for the BCPL programming language.

Run this code in your Python editor window by selecting '*Run/RunModule*'. It's likely it will now ask you to save the file, it's suggested you make a '*PAL*' folder in your drive (not Desktop!). You'll have to use this same process for all future pieces of code we run.

Starting from the top, we have simple comments that start with '`# text here`', where anything after the '`#`' is considered a comment. After this we have the '`print()`' function. Change this to print '`I enjoy this :)`'.

5 Exercise - Hello Loop

You've mastered printing to the console, now it's time to get the computer to do the boring repetitive tasks, starting with looping (the act of repeating). Open '*Resources/Week1/helloloop*' It's important to understand that looping is a powerful tool but shouldn't be used for everything. Run the program to see what it does.

In this program we have two types of loop, a '`for`' loop and a '`while`' loop. The difference here is that typically the '`for`' loop will be used for looking through a list of some type, i.e. 1, 2, 3..., whilst the '`while`' loop looks for a reason to stop. Both are useful depending on what you're doing. If you look at the line '`number = 0`', you can see that the variable (a piece of named computer memory) is given the value zero. What do you think '`number += 1`' does? What happens when you change the '`1`' to a '`2`'?

What happens when you add a zero to the end of the '`3`' and '`2`'? What happens when you keep adding zeros? Is there a limit and if so what does it tell you? How has the speed been affected?

Next, add a line under the '`print("Hello")`' and print '`"Jim"`'. Now add a line under '`"World"`' to say '`"is great!"`'. Be careful to keep the formatting to four spaces!

6 Exercise - Operators

Doing operations on numbers is essentially what computers do rapidly and flawlessly for humans which is what makes them so useful. Become familiar with these operations, they'll be part of your bread and butter in future programming sessions. Try assigning a calculation to a variable named '`myvar`' and printing it.

You may remember from GCSE maths that brackets make a difference to the order in which the Math is done. $(6-2)/2 = 2$ is not the same as $6-(2/2) = 5$ and different from $6(-2/2) = -6$. Replicate these sums in Python and print them to the console.

7 Resources & Further Reading

'*http://homepages.herts.ac.uk/~db12aba/*' – All content from these sessions updated weekly.

'*http://code.org/*' – A good resource testing your programming skills.

'*http://stackoverflow.com/*' – Highly recommended online help for programmers (NOTE: Employers are interested to know whether you're an active member of this site!).