SCIE1000, Computing Tutorial Week 1, Semester 1, 2011.

Aims

The aims of this introductory computing tutorial are:

- To gain some experience with using the computer laboratories (logging in, running software applications)
- To be able to run the Python software and use `IDLE`, the Python editor
- To write and execute Python programs that print text to the screen and perform simple calculations

1 Overview

[Ref: Appendix A of the Lecture notes, Section A.1, Pages 389–390.]

In this computing tutorial we will gain some basic experience with the Python programming language. No previous programming experience is assumed and you are encouraged to ask as many questions as you like from your tutors. At the start of most Sections of these tutorials you will find references to the lecture notes, the Python notes, or both. *You will find that if you have read this material (prior to attending your tutorial session) then completing the tutorial questions will be significantly easier.*

(a) Start by logging in to your computer. Run the Python program from the Start Menu and selecting Programs → ActiveState Active Python 2.5 → idle. You should see a window with a prompt for typing commands. If you have any difficulties logging in or running the program, please ask a tutor for help.

2 The Main Window and the Editor

[Ref: Appendix A of the Lecture notes, Section A.2.1]

It is possible to execute Python statements by typing directly into the main Python window

(a) Try typing “2 + 2” into the main window and press Enter.

However it is typically much more useful to write programs that can be saved into files and executed repeatedly without retyping. `IDLE` provides an editor that we will use for writing programs in SCIE1000.

Read through Section A.2.1 of the Python Notes and follow the steps to:

(b) Open an Editor window

(c) Write a simple Python program by typing the line

```
print 'Hello World!'
```

into the editor

(d) Create a Folder in your account to store your SCIE1000 programs (e.g. called “scie1000python”)

(e) Save your "Hello World" program into this folder, giving it an appropriate filename (e.g. "`helloworld.py`")

(f) Run your program from your editor window

3 Comments and Modules in Python Programs

[Ref: Appendix A of the Lecture notes, Sections A.2.2, A.2.3]

Read the relevant sections of the lecture notes. Modify your Hello World program from above so that:

(a) The first two lines of the program import standard modules into your program (see Section A.2.3 of the notes)

(b) You have a single line comment that describes what your program does (see Section A.2.2 and Program A.1 in the Python notes)

(c) Save and run your modified program again and check that it still produces the same output in the main window.

Note that while this simple program worked without importing modules, you should include these lines in every program that you write in SCIE1000. As for comments, you will see the benefits of adding these to programs later (when you write longer programs). Comments help document your program and can help you fix problems when your program doesn't work as expected.

4 Printing and Simple Calculations in Python

[Ref: Appendix A of the Lecture notes, Sections A.2.4, A.2.5]

(Note: you should open a new Editor window for each new program that you write in SCIE1000. You can save and load programs in the editor using the menus and mouse in much the same way as in a program like Microsoft Word).

Write a Python program (similar to Program A.2 in the Python notes) to perform the following calculations and print the results to the Main Python window:

(a) $2 + 2$

(b) $2312/11$

(c) 2^8

(d) 10^0

(e) Square root of 576 (i.e. $576^{0.5}$)

5 Displaying Numbers and Scientific Notation

[Refs: Appendix A of the Lecture notes, Section A.2.6]

By default, Python displays numbers in standard form, but switches to scientific notation for numbers that are large or small.

(a) Write a Python program (starting with the standard module import lines discussed above and a comment line to say what the program does) containing the following commands:

```
print 1e3
print 0.0000000001
```

(b) By modifying your program, determine at what order of magnitude Python switches to scientific notation for large numbers and for small numbers.

6 What Does it Do?

Consider the following Python statements:

```
print 6+4*5
print 'Viva Las Vegas!'
print 'I ate ',3+7,' apples.'
# print 'SCIE1000 is fun.'
print 2*3, 2*4, 2*5
print 0.00001, '0.00001', 2e0, '2e0'
```

(a) On a piece of paper, write down *exactly* what you expect the output of this program would be

(b) Type the program into the Editor and verify whether your answer to (a) was correct or not. If not, can you understand why? Ask a tutor if you need help.