Python Programming I
Lab

Objectives

- Explain the purpose of scripting languages and Python
- Explore basic syntax of Python, compare to C
- Use variables, expressions and statements in Python
- Explain function definitions and function calls (pure and with return value)

Table of Contents

Background Reading

Textbook:

How to Think Like a Computer Scientist
Download under the GNU Free Documentation License using following link:
www.greenteapress.com/thinkpython/thinkCSpy.pdf

Read chapters 1 to 5.

Why Python?

Both C and Python are high level languages, both need translation into machine code to execute. C is compiled, Python is interpreted. Describe the difference (draw a simple diagram of the path from source code to program execution and output):

C program	Python program

You have learned how to program in C. Simple program in C:

```
#include <stdio.h>

void main (int argc, char **argv)
{
     printf (“Hello World!\n”);
}
```

Write equivalent program in Python:

Instructor Sign-off: □

Notes common to all lab and home assignment problems

For every lab and home assignment, all work should go into your personal repository, subdirectory named mXX, where XX stands for the module number. For each problem, carefully name the program as described. The programs are extracted from your repository by a Python script, and errors in the program name will result in the instructor never seeing your program, and your mark for it will be ZERO!

There are always many ways how to solve a programming problem, and usually one or two ways which are fast, compact and elegant. E.g., Problem 3 can be solved in 5 lines of code.

Make sure to push your work to the server often, and have pushed the working version of the program by the deadline specified. The script extracting your programs from your repository will be run at any time after the deadline.

Problem 1

Write Python program named **`m02p01.py`** (module 2 problem 1), which will output single line, consisting of year, month, date, your sait email address, and your first and last name, all separated by single space, using the following format (substitute your own name and email for “john smith”!):

```
2016-01-31 john.smith@edu.sait.ca john smith
```

Problem 2

Write Python program named **`m02p02.py`**, which will output single line, which will show the host name of your computer, and how much memory is installed on your computer (hint: look for MemTotal in pseudo-file /proc/meminfo):

```
Hostname: SuperLinux, Memory: XXXX MB
```

Problem 3

Python has 29 keywords. Write Python program named **m02p03.py**, which will output the list of python keywords, one per line, sorted alphabetically, and then the same list, sorted alphabetically in reverse.

```
and
as
…
yield
yield
…
as
and
```

Problem 4

Write Python program named **m02p04.py**, which will take two numbers as parameters, and one of the operator words “plus”. “minus”, “times”, and print the result of the operation, e.g.,

```
m02p04.py 5 6 times
```

will print

```
5 x 6 = 30
```

Problem 5

Rewrite the program of previous problem such that the user is prompted for the two numbers after the program starts:

```
m02p05.py
First Number: 5
Second Number: 6
Operation: times
5 x 6 = 30
```

Problem 6

Write the program of problem 4 so it works like calculator, such that the user is prompted for the two numbers after the program starts, and continues repetitively until the word “exit” is typed:

```
m02p06.py
Calc: 5 6 times
5 x 6 = 30
```

```
Calc: 3 4 times
3 x 4 = 12
Calc: 7 8 times
7 x 8 = 56
Calc: exit
```

Problem 7

Write Python program named **m02p07.py**, which will print a list of all users on your linux system which have a home directory of the format /home/username, and print a pretty table, containing the user name using 20 characters and aligned left, and the home directory (hint: make use of file /etc/passwd, the table borders are made up from + sign, dash, and vertical bar):

```
m02p07.py
+-----------+---------------------------+
| user      | /home/user                |
| fred      | /home/fred                |
+-----------+---------------------------+
```

Write a function named `print_user_line(username)` to print each individual user line. The function will print the table top or bottom border line if the username is `None`.

Problem 8

Install python module pyutmp:

```
sudo pip install pyutmp
```

Write Python program **m02p08.py** which will print all user log-in/log-out information for last 24 hours, in the date, user, where from format, sorted in descending time order:

```
2016-11-07 14:57:39 fred /dev/pts/24
2016-11-04 16:34:59 bob server.sait.ca
2016-10-02 01:03:45 alice somewhere.google.com
```