Engineering & Computer Science
Department of Computer Science

CS161 Introduction to Programming and Problem Solving

Winter Term 2014

Due January 14

Homework #1

Exercise 1 is intended to introduce you to the programming environments you may encounter in developing Python code. It will require that you set up a Python environment on your own machine (if you have one) as well as remotely accessing a Linux host at the shell level.

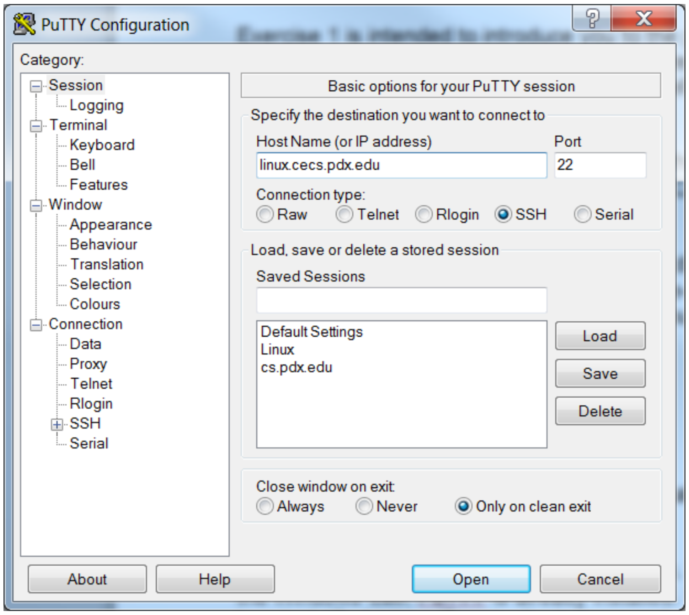
Step 1. Install the IDLE environment and Python on your home machine, laptop, etc. If you don't own your own computer, use the PCs in the CS Windows Lab in FAB 88 or EB 325 for the next part of the assignment.

Step 2. Write a Python program and save it (don't just use the Interactive Mode) under the name hw1-1_LastName-FirstName-PC.py or hw1-1_LastName-FirstName-Mac.py. When run, the program should display your name, e-mail address, the type of environment you are using and your major in the following format (but with your own information) using four Python `print()` statements:

```
Mark Morrissey
markem@pdx.edu
Windows Vista
Computer Science
```

The display should be left justified as shown above. Nothing else, just these four lines, in this order.

Step 3. If you are using a Windows machine, go to www.putty.org then download and install PuTTY. If you are using the PCs in the CS Windows Lab, PuTTY is already installed. Use PuTTY, login to linux.cecs.pdx.edu



Host Name (or IP address): linux.cecs.pdx.edu
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions:
Default Settings
Linux
cs.pdx.edu

Close window on exit: ○ Always ○ Never ● Only on clean exit

You will be prompted for your CS username and password (get this from the tutors in FAB 88).

If you are using a Mac, you can open a terminal (Go/Utilities/Terminal) window. In order to login, type

```
ssh yourLogin@linux.cecs.pdx.edu
```

You will be prompted for your password.

Step 4. Use the code editor **nano** to create pretty much the same program you wrote in Step 2 except rather than displaying “Windows” or “Mac” it will display “Linux.” To enter **nano**, type:

```
nano hw1-1_LastName-FirstName-Linux.py
```

key in your program and then type CTRL-c to exit **nano** (make sure to save the program when you are prompted).

This will save the code under the name hw1-1_LastName-FirstName-Linux.py. Run it to make sure that it works:

```
python3 hw1_LastName-FirstName-Linux.py
```

Step 5. E-mail the file you just created to yourself using the following command, where e-mailAddress represents your PSU computer login. You may know this as the Banner login.

```
mail e-mailAddress < hw1-1_LastName-FirstName-Linux.py
```

Make sure you logoff when you are done:

```
exit
```

Step 6. Save the e-mail you mailed yourself as hw1-1_LastName-FirstName-Linux.txt. The easiest way to do this is to open your PDX webmail by going to https://mail.pdx.edu/ and selecting inbox. You will find an e-mail from yourself listed as "[No Subject]" – open it and you will see the program you sent from your Linux account. When viewing the email, use the drop-down menu in the upper right of the displayed email and select “Show Original”. Save the email as a file on your desktop by right-clicking the original email and selecting “Save As”. When prompted, save it using the filename: hw1-1_LastName-FirstName-Linux.txt. Once you save it, if you open it, you will notice that it contains quite a bit more than just your program code. It also contains the e-mail headers inserted by the various e-mail programs that it went through to get it from the Linux command line to your e-mail reader":

```
Delivered-To: markem@pdx.edu
Received: by 10.112.125.42 with SMTP id mn10csp3857733lbb;
        Fri, 2 Jan 2015 10:24:08 -0800 (PST)
X-Received: by 10.66.118.168 with SMTP id kn8mr59250527pab.45.1420223046881;
        Fri, 02 Jan 2015 10:24:06 -0800 (PST)
Return-Path: markem@cs.pdx.edu

… <a lot of stuff> …

Message-Id: <20150102182259.3D6884C148B@ada.cs.pdx.edu>
From: markem@cs.pdx.edu (Mark Morrissey)
X-Spam-Status: No, score=2.4 required=5.0 tests=BAYES_00,L_RCVDUSER,
      MISSING_SUBJECT,RCVD_NOT_IN_IPREPDNS,TVD_SPACE_RATIO,T_RP_MATCHES_RCVD
      autolearn=no version=3.3.2
X-Spam-Level: **
X-Spam-Checker-Version: SpamAssassin 3.3.2 (2011-06-06) on expn.cat.pdx.edu
X-Virus-Scanned: clamav-milter 0.98.5 at expn
```

```
X-Virus-Status: Clean
X-PSU-Routing: default-relay

print("Mark Morrissey")
print("markem@cs.pdx.edu")
print("Linux")
print("Computer Science")
```

Step 7. Submit the two files, hw1-1_LastName-FirstName-Linux.txt and hw1-1_LastName-FirstName-PC.py using the CS161 D2L site. The filenames **MUST** be as specified.

Exercise 2 is designed to familiarize yourself with a simple python turtle graphics program and to afford you the opportunity to experiment with the turtle graphics system.

Step 1. On a Windows PC (your own or one in the lab) type in the following python program. Use the same naming scheme as exercise 1, but change 'hw1-1' to 'hw1-2'. When entering this program, also change the comment on the second line to your actual name and student ID. **Note: indentation matters in the Python programming language!** The indented potions of the code below are indented using either 4 or 8 spaces. Please do not use the tab key to indent as sometimes very subtle bugs can occur.

```
# turtle_example.py - example program for turtle graphics
# (c) 2015, First_Name Last_Name, PSU_Student_ID
#

import turtle

turtle.color('red', 'yellow')
turtle.begin_fill()
while True:
    turtle.forward(200)
    turtle.left(170)
    if abs(turtle.pos()) < 1:
        break
turtle.end_fill()
turtle.done()
```

Step 2. Run your program and study the output. We will discuss this program in class.

Step 3. Look at the turtle documentation at the URL https://docs.python.org/3/library/turtle.html#module-turtle and see if you can figure out what the commands that start with 'turtle.' are doing. Note that the documentation will leave off the 'turtle.' part.

Step 4. Step 7. Submit the file hw1-2_LastName-FirstName-PC.py using the CS161 D2L site.