

# Program Py - Python Exploration

[Submit Assignment](#)

---

**Due** Nov 11 by 11:59pm    **Points** 32    **Submitting** a file upload    **File Types** py  
**Available** Sep 30 at 12:01am - Nov 13 at 11:59pm about 2 months

---

## Program Py – CS 344

### Overview

In this assignment, you will be given ZERO instruction in HOW to accomplish it! The actual requirements are very simple to satisfy. This is a test of how well you can research the topic and satisfy the conditions of this Assignment on your own. This is how much of the real world works - you'll be thrown into situations that are full of unknowns, and you'll need to come up with a solution!

You are becoming a computer scientist, not a technician: you will learn how to handle ANY language thrown at you, not just the ones you are spoon-fed, by learning good research habits and techniques. The more problems you encounter and solve, the greater problem set you'll be able to infer solutions to from your background of knowledge.

This is by far the easiest assignment in this course - it's worth the least points, with the most time to accomplish it. You will use what you've learned here in later courses at OSU.

### Specifications

For this assignment, you will create a script in the Python language. You can read more about Python here:

<http://www.python.org/> [\(http://www.python.org/\)](http://www.python.org/)

All execution, compiling, and testing of this script should ONLY be done in the bash prompt on our class server!

Your script must satisfy the following requirements:

1. Be contained in one single file, called "mypython.py".
2. When executed, create 3 files in the same directory as your script, each named differently (the name of the files is up to you), which remain there after your script finishes executing. Each of these 3 files must contain exactly 10 random characters from the lowercase alphabet, with no spaces ("hoehdgwkdq", for example). The final (eleventh) character of each file MUST be a newline character. Additional runs of the script should not append to the files. Thus, running `wc` (wordcount) against your files in the following manner must return 11:

```
$ cat myfile
gkwjhcfcikf
$ wc -c myfile
11 myfile
```

3. When executed, output sent to stdout should first print out the contents of the 3 files it is creating in exactly the format given below.
4. After the file contents of all three files have been printed, print out two random integers each on a separate line (whose range is from 1 to 42, inclusive).
5. Finally, on the last (sixth) line, print out the product of the two numbers.

You do not have to parse and read the data back in from the files created in step 2 in order to complete step 3. For step 3, just dump the contents that you already randomly generated in your program directly onto the screen, if that's the easiest way for you. I recommend that you look into `sys.stdout.write()` for outputting text, as it will allow you to control newlines better.

The graders will simply be checking for the above requirements to assign your grade, using the exact format shown below. Further, no help on this assignment will be provided by the Instructor or TAs at any time.

You can choose to use Python 2 or Python 3. The TAs will try all three of these commands to find one that works with your script:

```
$ python mypython.py
$ python2.7 mypython.py
$ python3 mypython.py
```

Python is NOT a topic that will be covered on the Final.

## Example Output

Here's an example of the script being run. All output is random except for the last line (since it's the product of the 4th and 5th lines), and your output must match the format shown here:

```
$ python mypython.py
vwdzwuxwvh
nibqaackrp
pxeugdqnwc
8
10
80
```

## Hints

I HIGHLY recommend that you develop this program directly on our server, instead of on your personal machine. Doing so will prevent you from having problems transferring the program back and forth, which can cause compatibility issues.

If you do see ^M characters all over your files, try this command:

```
% dos2unix bustedFile
```

## What to Turn In and When

Please submit simply your Python script. As our Syllabus says, please be aware that neither the Instructor nor the TA(s) are alerted to comments added to the text boxes in Canvas that are alongside your assignment submissions, and they may not be seen. No notifications (email or otherwise) are sent out when these comments are added, so we aren't aware that you have added content! If you need to make a meta-comment about this assignment, please include it in a #comment near the top of the script itself, or email the person directly who will be grading it (see the [Home](#) page for grading responsibilities).

The due date given below is the last minute that this can be turned in for full credit. The "available until" date is NOT the due date, but instead closes off submissions for this assignment automatically once 48 hours past the due date has been reached, in accordance with our [Syllabus Grading policies](#).

## Grading

Note that this assignment is PASS/FAIL, meaning all of the points are awarded or not awarded based on whether or not you accomplish ALL of the requirements! If you do not accomplish all of them, you will receive a zero. But of course you'll accomplish them, because this Assignment is really easy! It's practically extra credit!

Remember that this will be graded only on os1, in a bash prompt. Make sure your code works there!

The TAs will use this exact set of instructions: [ProgramPy Grading.pdf](#)  to grade your submission.

Some Rubric				
Criteria	Ratings			Pts
Works	<b>32.0 pts</b> <b>Full Marks</b>	<b>28.8 pts</b> <b>1 day late</b>	<b>24.0 pts</b> <b>Two days late</b>	32.0 pts
				Total Points: 32.0