Video Response Questions: Debugging

[Submit Assignment](https://mchenry.instructure.com/courses/1423595/assignments/7792350?module_item_id=15506206)

* **Points** 15
* **Submitting** a text entry box or a file upload

**INSTRUCTIONS FOR THIS ASSIGNMENT:**

1. Log in to your free student account at Lynda.com using the link found [here](https://mchenry.instructure.com/courses/1423595/modules/2053665" \o "Login to lynda.com - must do this each time you login" \t "). [NOTE: If you step away from your computer for a while, your connection to Lynda.com may timeout. If this happens, just login again using the same link.]
2. Watch the linked video segment(s) on Lynda.com from the **Foundations of Programming: Fundamentals** series with Simon Allardice. Each video segment is about 5 minutes in length.
3. Briefly answer the question(s) for each video in your own words. Use complete sentences.

Video:**[Introduction to Debugging](http://www.lynda.com/Developer-Programming-Foundations-tutorials/Introduction-debugging/83603/90479-4.html" \t "_blank)**

(1) Explain the difference between syntax errors and logic (semantic) errors. Tell how each is detected. (5pts). Syntax errors are a kind of format error, a missing colon or semi-colon, a capitalized letter where it should be lower case, you could think of them as sentence errors. Then there are Logic errors where the Syntax is good but there is a flaw in the formula/organization. Example would be like trying to divide any number by 0, or maybe one of your loops doesn’t go round enough times or too many times.

-------------------------------------------------

Video:**[Tracing Through a Section of Code](http://www.lynda.com/Developer-Programming-Foundations-tutorials/Tracing-through-section-code/83603/90480-4.html" \t "_blank)**

(2) Explain the concept of tracing and how you might do something similar in Python. (5 pts)

Low Tech solutions “Tracing.”

After reproducing the problem and getting a sense of where it might be, a programmer can insert Trace Message line where he thinks the problem exists and run the program. The goal is to follow the flow of what the program is supposed to do and narrow down the section of code where the error is by checking for the Trace inserts, they allow you to see how far the program is getting before it crashes. It appears to be a very tedious and time-consuming process; however, it does have the advantage of not requiring any special tools to run. For an advanced programmer this might be an option. As a beginner I might need the help of a debugger.

(3) Explain how breakpoints work and what it means to step through your code. How might this benefit you as a programmer? (5 pts)

Reproduce the problem, hopefully the same problem and does it occur at the same time in the same way?

A Break Point is a specific point in the program where a Stop sign is intuitively inserted allowing the programmer to jump in and jump out of the program (checking and tweeking) while it is running. The debugger lets us see variables and their values in real time while also highlighting possible errors. Break Points can be set, deleted, and reset as many times as necessary.

*For a quick overview of using the debugger in PyCharm Edu, see the [PyCharm Debugger Quick Reference](https://mchenry.instructure.com/courses/1423595/pages/pycharm-debugger-quick-reference" \o "PyCharm Debugger Quick Reference" \t ").*

*\*Examples of how certain Javascript statements used in the video might appear in Python. Notice that the alert() function in Javascript creates a popup message because the script is running in a webpage. But the concept is just to display a message to the user which we might do with a print statement in Python.*

|  |  |
| --- | --- |
| Javascript | Python |
| function startCountdown() { | def startCountDown(): |
| // check if not a number | # check if not a number |
| if (isNaN(minutes)) { | if not isnumeric(minutes): |
| var a = 100; | a = 100 |
| alert(result); | print result |

Rubric

**SomVideo Response Questions: Debugginge Rubric**

| SomVideo Response Questions: Debugginge Rubric | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| Introduction to Debugging | |  |  | | --- | --- | | Full Marks  5.0 pts | No Marks  0.0 pts | | 5.0 pts |
| Tracing Through a Section of Code | |  |  | | --- | --- | | Full Marks  5.0 pts | No Marks  0.0 pts | | 5.0 pts |
| Using Debuggers | |  |  | | --- | --- | | Full Marks  5.0 pts | No Marks  0.0 pts | | 5.0 pts |
| Total Points: 15.0 | | |

[Previous](https://mchenry.instructure.com/courses/1423595/modules/items/15506205)[Next](https://mchenry.instructure.com/courses/1423595/modules/items/15506208)