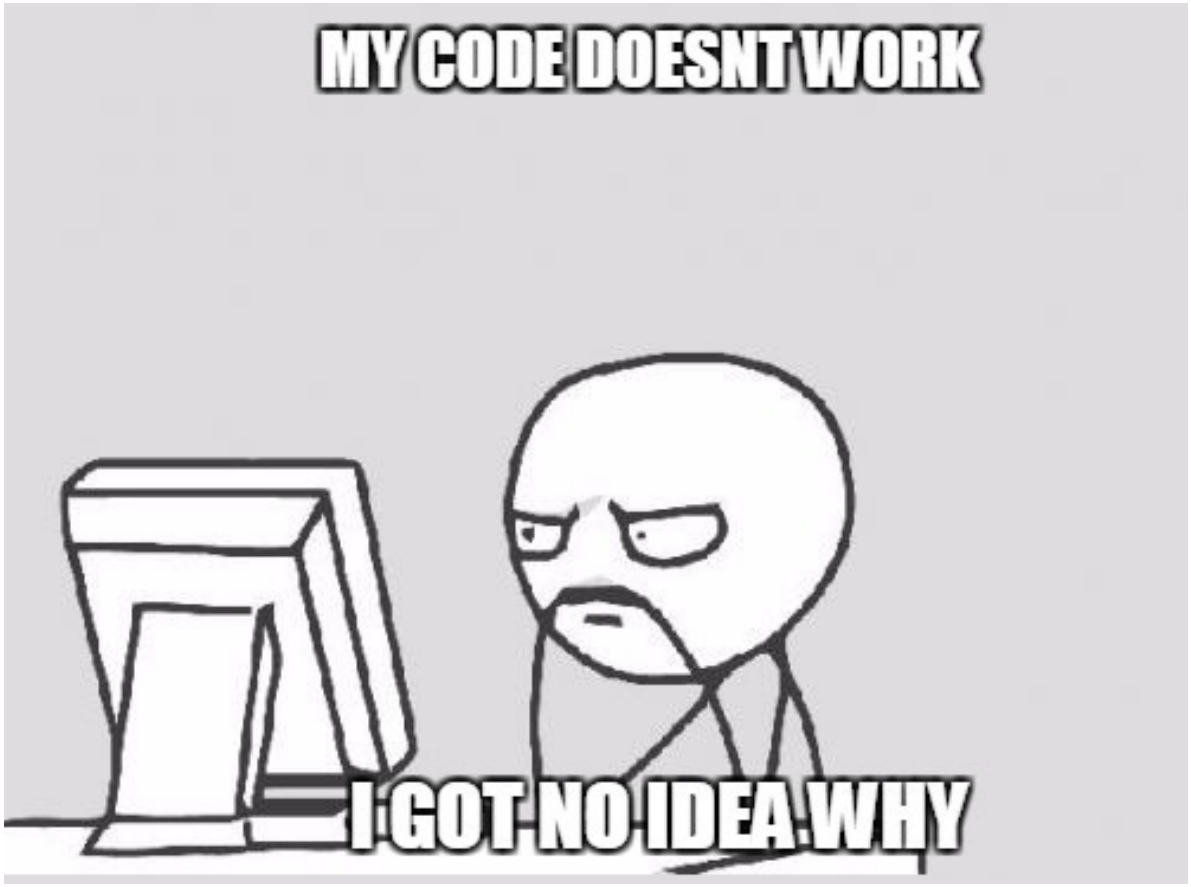
Lecture 11: Debugging and Testing

CS 51P

October 11, 2023

Announcements

- Checkpoint 2
  - Functions through Debugging and Testing
  - October 23, 2023


MY CODE DOESNT WORK
I GOT NO IDEA WHY

Common Types of Errors

- **Syntax Errors:** there is something wrong with the structure of the program, and Python doesn't understand it
- **Runtime Errors:** something goes wrong while the program is running
- **Semantic Errors:** the program runs, but it doesn't do what you want it to do

Handling Syntax Errors

1. Find the bug

```
/Users/eleanor/PycharmProjects/51P/2019fa/venv/bin/python /Users/eleanor/Py
  File "/Users/eleanor/PycharmProjects/51P/2019fa/Demos/demo05.py", line 1
    print("*"*80"\n")
                 ^
SyntaxError: invalid syntax

Process finished with exit code 1
```

2. Do you see the problem?
    1. If yes, fix it!
    2. If no, try running through the list of common syntax bugs
    3. If still no, check your class notes, discuss the problem abstractly with a friend ("what's the right syntax for…"), or ask a TA/instructor (it's ok to get help!)

Common Syntax Errors

- Misspelling a variable name or a function name
- Missing quotation marks around a string
- Mismatched parentheses or quotation marks
- Missing a colon at the end of an if/while/for statement
- Using = instead of ==
- Using a Python keyword as a variable name

Make sure you remembered to save your file
after making your changes!

Example

```
in = int(input("Pick a number\n"))
if in = 13:
    print("I am also fond of the number 13!")
elif in > 13:
    print("I am fond of the number 13, which is "
          + str(in-13) + " less than " + str(in)
else
    print("I am fond of the number 13, which is "
          + str(13-in) + " more than " + str(in)
in2 = input("Do you like tea?)
while in2 != "yes" and != "no":
    in2 = input("Please answer yes or no. Do you like tea?")
if in2 == "yes":
    print("Great!")
else:
    print("That's too bad.")
print("Bye!)
```

SyntaxError
SyntaxError
SyntaxError
SyntaxError
SyntaxError
SyntaxError
SyntaxError

Can you find the
the **mistake**?

1 2 3 4 5 6 7 8 9

Handling Runtime Errors: Program Hangs

- You are probably in an infinite loop!
- Add print statements to figure out how far you got
- Add print statements to find line(s) that repeat over and over

- Your program might also just be waiting for an input

Handling Runtime Errors: Exceptions

- NameError: Python doesn't recognize a (variable) name
  - Find the bug!
  - Did you forget quotation marks around a string?
  - Did you misspell a variable name? Make a typo?
  - Is the variable you are trying to use in scope? Use before define?

Scope

```
def good_choice(num):
1    b = (num == fav)
2    return b

def main():
1    fav = 47
2    number = int(input())
3    if good_choice(number):
4      print("yay")
5    else:
6      print("boo")
```

Storing a value in a variable:

1. If there is a variable with that name in the current function's stack frame, store the value in that variable
2. Otherwise create a new variable in the current function's stack frame and store the value there

Using a variable:

1. Check for a local variable with that name. If it exists, use the value stored in that variable
2. Otherwise get a NameError

Exercise

```
def print_example(s4,s5):
    s1 = 3*s4
    s2 = s4+s5
    print(s1)
    print(s2)
    return s1+s2

s1 = '!'
s2 = '?'
print(s1)
s3 = print_example(s1,s2)
print(s2)
print(s3)
print(s4)
```

Handling Runtime Errors: Exceptions

- NameError: Python doesn't recognize a (variable) name
  - Find the bug!
  - Did you forget quotation marks around a string?
  - Did you misspell a variable name? Make a typo?
  - Is the variable you are trying to use in scope? Use before define?
- TypeError: Python can't perform that operation/function on that type
  - Find the bug!
  - Are the types that the error reports the type you expected?
  - Add a print statement on the previous line and print out all the variables/values on that line. Are they what you expect?
- ValueError: Python can't perform that operation/function on that value
  - Find the bug!
  - Add a print statement on the previous line and print out all the variables/values on that line. Are they what you expect?

When your code runs…

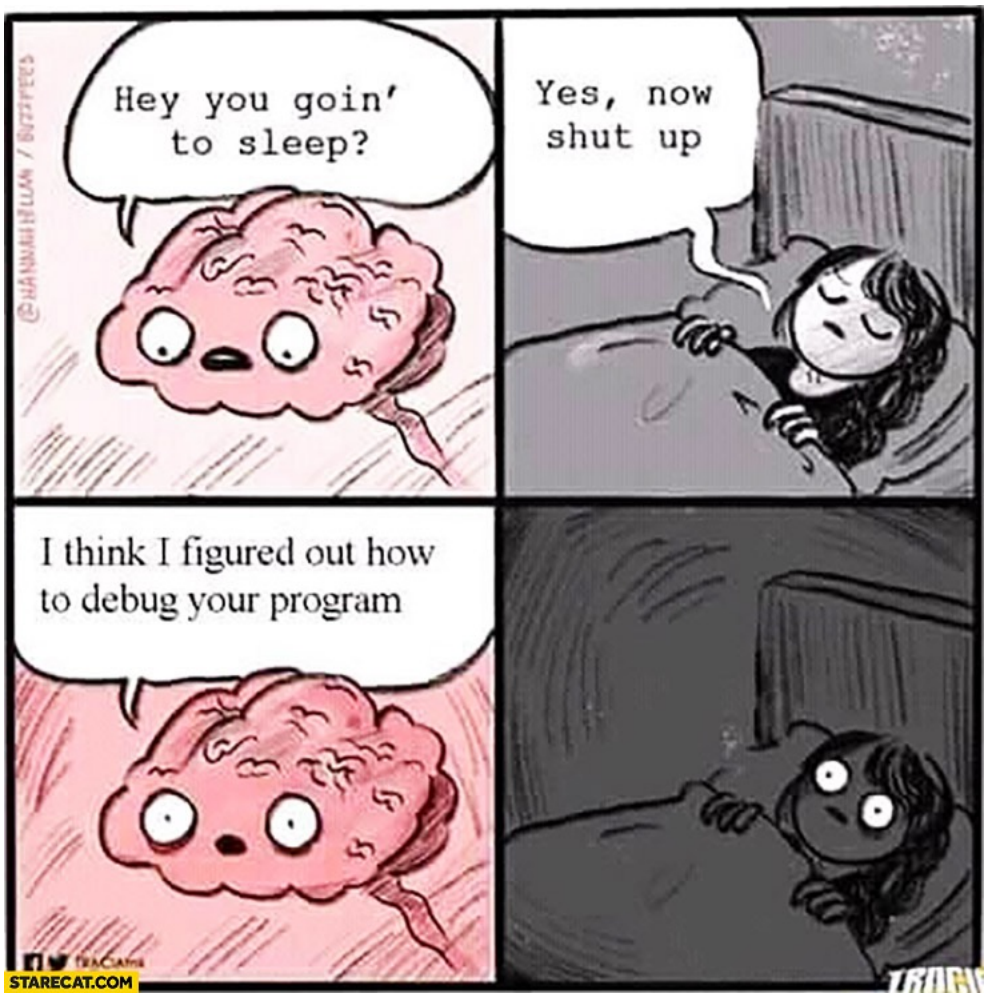
Example 4

- Simple password checking:
  - Password should contain at least 8 characters
  - Only one requirement

Handling Semantic Errors

- Add print statement in the appropriate places
- Print out the value of variables that you want to keep track of
- Compare the print out result with your expected result

Take a break


Hey you goin' to sleep?
Yes, now shut up
I think I figured out how to debug your program
STARECAT.COM

When your code runs…

Testing

- Try running your function with different values, called test cases, and make sure it returns the right value

- Branch Testing (white-box testing)
  - make sure that every line of code is executed by at least once
  - for conditionals, try include a test case that makes the condition evaluate to True and a test case that makes the condition evaluate to False
  - for loops, try to include test cases that make the program go through the loop 0 times, 1 time, and lots of times

- Corner-Case Testing (black-box testing)
  - include the "weird" values in your test cases
  - e.g., for ints, include negative numbers and zero, as well as positive
  - e.g., for strings, include the empty string

Testing in Python

- Create a new file called <program_name>_tester.py
- Import the functions you want to test

```
from demo11 import sum_even
```

- Using assert statements to test program behavior

```
assert <condition>
```

Example

demo11.py

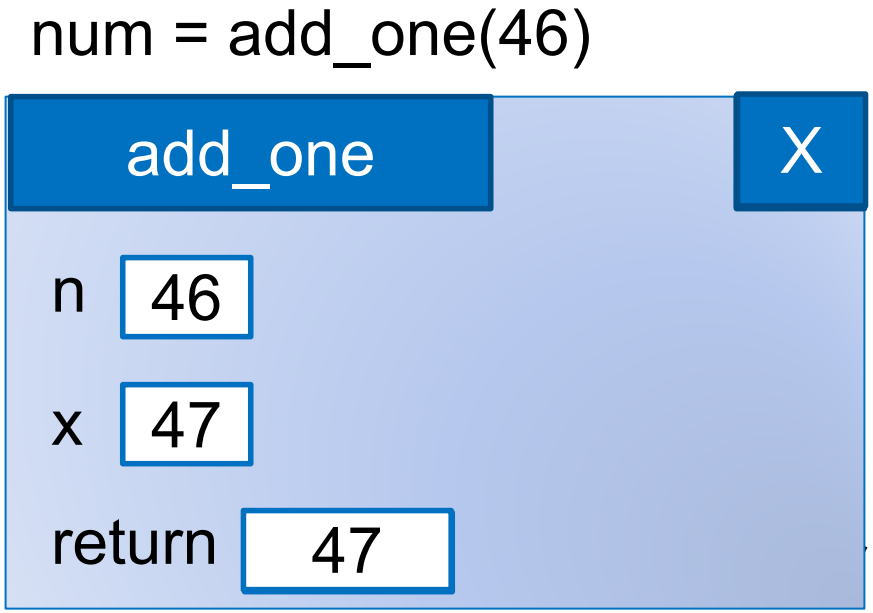
```
def sum_even(start, end):
  """
  Computes the sum of the
  even numbers between <start>
  and <end> (inclusive).
  :param start: (int) one end
                      of range
  :param end: (int) other end
                     of range
  :return: (int) sum of evens
  """
  for i in range(start, end):
    if i % 2 == 0:
      sum = i
```

demo11_tester.py

```
from demo11 import sum_even


def main():
  assert type(sum_even(1,5)) == int
  assert sum_even(1,5) == 6
  assert sum_even(1,6) == 12
  assert sum_even(2,5) == 6
  assert sum_even(2,6) == 12
  assert sum_even(1,1) == 0
  assert sum_even(2,2) == 2



if __name__ == "__main__":
  main()
```

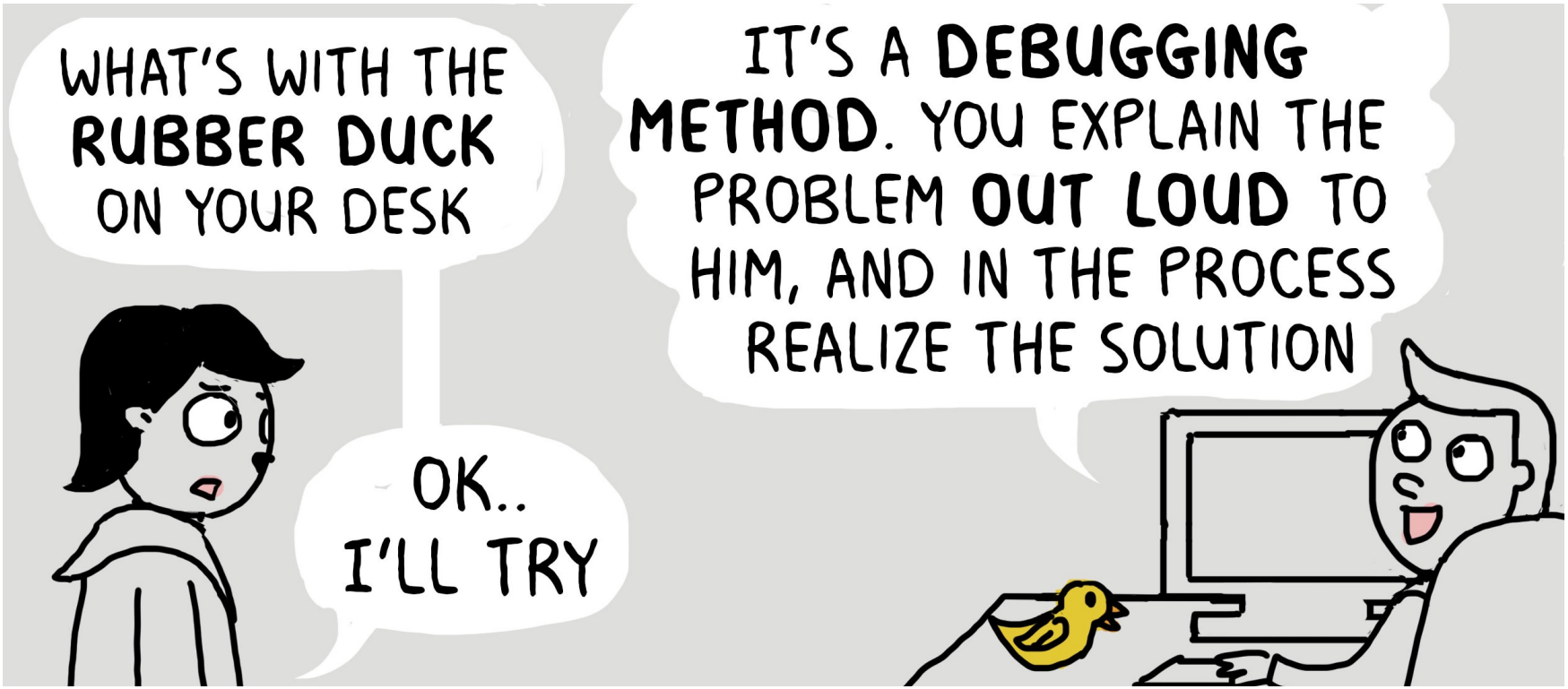
Code Tracing

- Execute the program line by line by hand



- If you get the right answer by hand, add print statements to determine where your code starts doing something different

Rubber-Duck Debugging



Debugging…

Debugging…


YOU HAVE DISHONORED ME FOR TOO LONG.
EVEN THOUGH FOR EVERY ONE OF YOU I STRIKE DOWN, ANOTHER TWO MAY TAKE HIS PLACE...
... I WILL KILL YOU ALL!
AAAAAAAAA
AAAAA
DEBUGGING IS A STATE OF MIND.
WWW.NIKTALK.COM
© 2000 BARRY T. SMITH

Bonus Exercise

- Example 2

- Example 5
  - sum_even again!