CIS 122 Fall 2015 Project 8 debug files, find and fix errors
Due Wednesday, March 9, 11:59 PM

Introduction to debugging

Everyone who programs must learn to debug programs.

There are 3 major categories of bugs:

1. Violation of Python language rules

Example

```
if x = 3:
```

This should have a double equals == to compare x to 3.
Python gives you error messages about rule violations.

```
print("Hello Jane"
number = 6
```

Python complains of a syntax error here, but it really is caused by an error on the previous statement.

2. Errors that occur when running your program

```
option = "b"
print(options)
```

When run, gets undefined variable **options**

3. Program runs without obvious errors but does not solve the problem it was designed to solve.

The Mars Observer radar unit correctly measured the height of the satellite above Mars and sent the number of **feet** above Mars to the altitude adjustment module which expected to get the number of **meters**.

In this project, you will run programs, note the errors and correct them.

We'll start with a very simple program with just one error

error0.py

```
# error0.py

capital = 'Salem'
state_name = 'Oregon'

print(state_name, capitol)


# when run, gets this error message
# NameError: name 'capitol' is not defined
```

Identify the fixes you make with a
#-- (brief description of your fix or fixes)

The fixed version:

```
# error0_FIXED.py

capital = 'Salem'
state_name = 'Oregon'

#--changed variable name from capitol to capital
print(state_name, capital)
```

Generally, if you find one error, there is a very good chance that there is another error (or more!) still present in your program.

Another example, that may have more than one error

```
# error1.py

def pesos_to_euros(n_pesos):
    ''' Given pesos, return equivalent number of euros'''
    rate = 50 # pesos per euro
    euros = n_pesos / rate
    return euros

# Ask for number of pesos, print amount of euros they will buy

euros = 0.02

hint = "Type number of pesos: "
pesos = int(input(hint))

pesos_to_euros(pesos)

print(pesos, 'pesos are worth', euros, 'euros')

# No error message, but always says "worth 0.02 euros"
```

You type 1000 for the number of pesos and the program announces that they are worth 0.02 euros.

You notice that the function gets called and appears to calculate correctly.

This looks like a possible fix.

```
#--Assigned result of function call to euros
euros = pesos_to_euros(pesos)
```

The program now calculates 50 pesos will be worth 1 euro.

What about 100.5 pesos? That gets an error. Maybe we should allow float values instead of int.

```
Type number of pesos: 50.77
50.77 pesos are worth 1.0154 euros
```

Rounding the number of euros would help.

Here is a fixed version of the program; it repairs the errors of the program.

```
# error1_FIXED.py

def pesos_to_euros(n_pesos):
    ''' Given pesos, return equivalent number of euros'''
    rate = 50 # pesos per euro
    euros = n_pesos / rate

    #--Added rounding
    euros = round(euros, 2)

    return euros

# Ask for number of pesos, print amount of euros they will buy

euros = 0.02

hint = "Type number of pesos: "

#--changed int to float (allows converting 6.95 pesos, for example)
pesos = float(input(hint))

#-- assigned result of function call to variable euros
euros = pesos_to_euros(pesos)

print(pesos, 'pesos are worth', euros, 'euros')
```

Bonus work on this example would be to introduce a while loop to allow changing a number of different peso amount to euros…

Except for an exercise, do not work with these examples, instead, work on the following programs.

P8_1_unending.py
5 points

Symptom: When I run this program, I have to keep adding names; I don't have an option to Quit.

Fix the program, save it as **P8_1_fixed.py**

Rubric

2 points **#--** tells what changes you made to fix the program
3 points Actual fixes correct the program.

```
# P8_1_unending.py

# Help - when I run this program it keeps going and going...


my_list = ['Jim', 'Ashley', 'Amanda', 'Zelda']

# Add more names

hint = "Type Q to quit or first name to add (such as Anne): "
name = input(hint).title()
while name != 'Q':
    my_list.append(name)
    print("Added", name)
```

Example of program output when fixed (includes print of list)

```
Type Q to quit or first name to add (such as Anne): Jordan
Added Jordan
Type Q to quit or first name to add (such as Anne): Melissa
Added Melissa
Type Q to quit or first name to add (such as Anne): Anne
Added Anne
Type Q to quit or first name to add (such as Anne): Q
Jim
Ashley
Amanda
Zelda
Jordan
Melissa
Anne
```

P8_2_oops.py
5 points

Symptom: When I run this program, I cannot find names like Amanda that are in the list.

Fix the program, save it as **P8_2_fixed.py**

Rubric

2 points **#--** tells what changes you made to fix the program

3 points Actual fixes correct the program.

```
# P8_2_oops.py
# Help - when I run this program it does not work right.
# I cannot find names like amanda that are in the list
#
def search(who, names):
    ''' Return index of name or None if not found'''
    index = 0
    for person in names:
        if person == who:
            return index
        else:
            return None

my_list = ['jim', 'ashley', 'amanda', 'zelda', 'brittany']
my_gpa =  [3.05, 4.01,      2.87,      4.20,     1.95]


hint = "Type Q or type a name to search for (such as Jim): "
find_me = input(hint).lower()

while find_me != 'Q':
    index = search(find_me, my_list)
    if index == None:
        print(find_me, 'NOT found in names list')
    else:
        print(find_me, 'found at offset', index, 'in names list')
        print(find_me, 'has a', my_gpa[index], 'grade point average')
    find_me = input(hint).lower()
```

Example of minimal fix (5 points)

```
Type Q or type a name to search for (such as Jim): zelda
zelda found at offset 3 in names list
zelda has a 4.2 grade point average
Type Q or type a name to search for (such as Jim): amanda
amanda found at offset 2 in names list
amanda has a 2.87 grade point average
Type Q or type a name to search for (such as Jim): Q
```

Example of better fix (5 points + 1 bonus)

```
Type Q or type a name to search for (such as Jim): amanda

Amanda found at offset 2 in names list
Amanda has a 2.87 grade point average
Type Q or type a name to search for (such as Jim): zelda

Zelda found at offset 3 in names list
Zelda has a 4.2 grade point average
Type Q or type a name to search for (such as Jim): q
>>>
```

P8_3_fixme.py

10 points

Symptom: When I run this program, I cannot find names like Amanda that are in the list.

Fix the program, save it as **P8_3_fixed.py**

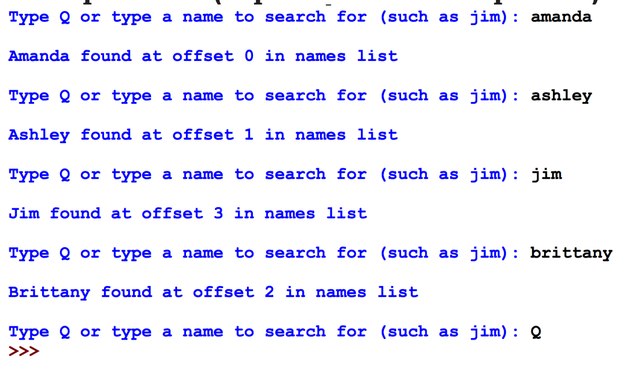
Rubric
4 points
#-- identifies each of several flaws in the program and tells what you did to fix each one.

6 points A series of actual fixes correct the program.

```
# P8_3_fixme.py

# Help - when I run this program it does work right.
# It does not find names like ashley in the names list
#

def linear_search(who, names):
    ''' Return index of name or None if not found'''
    index = 0
    for person in names:
        if person == who:
            return index
        elif person > who:
            return None
    return None

names = ['jim', 'tom', 'ashley', 'amanda', 'brittany', 'megan']

hint = "Type Q or type a name to search for (such as jim): "
find_name = input(hint).lower()

while find_name != 'Q':
    index = linear_search(find_name, names)
    print()
    if index == None:
        print(find_name, 'NOT found in names list')
    else:
        print(find_name, 'found at offset', index, 'in names list')
    print()

    find_name = input(hint).lower()
```

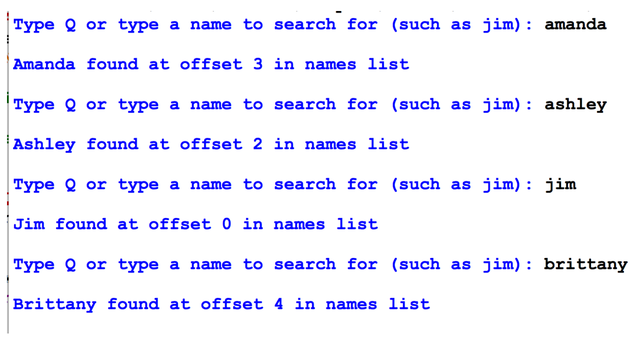
There may be several strategies for fixing this program.
Clearly identify what strategy you are using to fix the program.
Then test your fixed program to make sure it works correctly.

Example of fix (capital names are optional)



```
Type Q or type a name to search for (such as jim): amanda

Amanda found at offset 0 in names list

Type Q or type a name to search for (such as jim): ashley

Ashley found at offset 1 in names list

Type Q or type a name to search for (such as jim): jim

Jim found at offset 3 in names list

Type Q or type a name to search for (such as jim): brittany

Brittany found at offset 2 in names list

Type Q or type a name to search for (such as jim): Q
>>>
```

Another solution gives this:



```
Type Q or type a name to search for (such as jim): amanda

Amanda found at offset 3 in names list

Type Q or type a name to search for (such as jim): ashley

Ashley found at offset 2 in names list

Type Q or type a name to search for (such as jim): jim

Jim found at offset 0 in names list

Type Q or type a name to search for (such as jim): brittany

Brittany found at offset 4 in names list
```

P8_4.py

5 points

Symptom: When I run this program, I get a syntax error.

Fix the program, save it as **P8_4_fixed.py**

Rubric

2 points **#--** tells what changes you made to fix the program
3 points Actual fixes correct the program.

```
# P8_4.py

# Display options, do the right thing, repeat

def get_option(my_codes, my_descriptions):
    ''' Display codes and descriptions from lists
        Ask user to choose a code
        Return user choice as single lower-case character
    '''
    for count in range(len(my_codes)):
        print(my_codes[count], my_descriptions[count])

    answer = input("Type your choice: ")
    option = answer[0].lower()
    for code in my_codes:
        if code[0].lower() == option:
            return option
    print("\nPlease try again")
    return get_option(my_codes, my_description)

codes = ['Q', 'B', 'S', 'P']
descr = ['Quit', 'Buy', 'Sell', 'Panic']

while True == True:
    choices = get_option(my_codes, my_descriptions)
    if choice == 'Q':
        break
    elif choice = 'B':
        print("Buy more and save even more!")
    elif choice = 'S':
        print('Sell and earn big money now!')
    elif choice = 'P':
        print('The yen is down, oil is down, abandon all hope - PANIC')

print("Done")
```

Example of fix

```
q Quit
b Buy
s Sell
p Panic
Type your choice: b
Buy more and save even more!
q Quit
b Buy
s Sell
p Panic
Type your choice: s
Sell and earn big money now!
q Quit
b Buy
s Sell
p Panic
Type your choice: q
Done
```