Exam Review

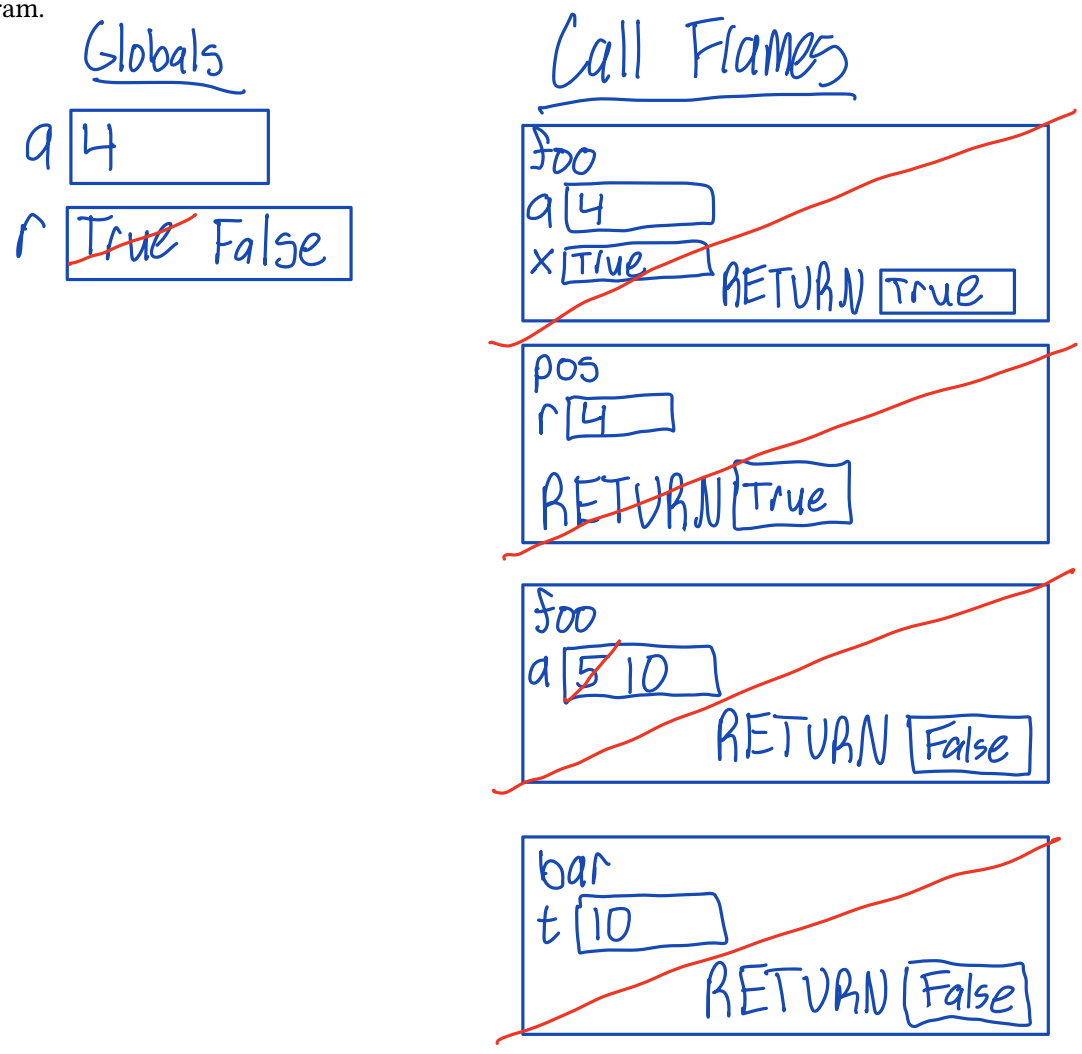
CS 1109, 2023SU

July 20, 2023

Problem 1: Tracing

Below is a snippet of Python code. Draw the memory diagram (global space and call frames) that corresponds to executing the program.

```
1  def foo(a):
2      if a % 2 == 1:
3          a = 2 * a
4          return bar(a)
5      else:
6          x = pos(a)
7          return x
8
9  def bar(t):
10     return t % 4 == 1
11
12 def pos(r):
13     if r > 0:
14         return True
15     return False
16
17 a = 4
18 r = foo(a)
19 r = foo(a + 1) and r
```



Now consider the following `for`-loop:

```
for i in range(1, 21):
    if i % 2 == 0:
        print("Even!")
    else:
        print("Odd!")
    print(i)
```

How many lines are printed by the above loop?

$2 \cdot 20 = \boxed{40}$

(the loop performs 20 iterations, and each iteration prints 2 lines)

Write below the first *six* lines of printed output that the above code snipped produces.

Odd!
1
Even!
2
Odd!
3

Problem 2: Coding Practice

Shown here are two function headers with complete specifications but empty bodies. It is your task to *implement* these functions according to their specification. Do not modify the headers or specifications of the given functions.

```
# p2.py

def is_vowel(c):
    """
    Returns True if c is a string of length one and is a
    vowel (uppercase or lowercase), and False otherwise.

    'y' or 'Y' are considered consonants, not vowels.

    Precondition c: c is a string
    """
    return len(c)==1 and c in 'aeiouAEIOU'

                    OR

    return c=='a' or c=='A' or c=='e' or c=='E' or c=='i' or c=='I'
           or c=='o' or c=='O' or c=='u' or c=='U'


    def more_vowels(s):
    """
    Returns True if s contains more characters that are vowels than
    characters that are not vowels, and returns False otherwise. In
    the case of a tie, this function returns False.

    Precondition: s is a string.
    """
    vowels = 0
    others = 0
    for c in s:
        if is_vowel(c):
            vowels = vowels + 1
        else:
            others = others + 1
    return vowels > others
```

Problem 3: Fill In The Blank

In the table below, write the *value* that the corresponding expression evaluates to. If an error is thrown while evaluating an expression, write "**Error**". If nothing is returned by an expression, write `None`.

Expression	Value
`3 + 4.1`	7.1
`2.0 - 1`	1.0
`int(4.6)`	4
`float('2')`	2.0
`3 + '4'`	error
`'55' + "4"`	'554' or "554"
`5 // 2`	2
`'a' * 3.1`	error
`'animals'[3]`	'm'
`'animals'[1:]`	'nimals'
`'animals'[5:8]`	'ls'

Consider the function `foo` defined below. Complete the below table in the same manner as the previous one, assuming you have access to the function `foo`.

```
def foo(g):
    if g > 1 or g < -2:
        g = abs(g)
        g = g - 2
    elif g == 0:
        return 1
    else:
        g = 42
    return g
```

Expression	Value
`foo(0.5)`	42
`foo(0)`	1
`int(foo(-2.4))`	0
`foo(int(0.8))`	1
`foo(-2.3)`	0.3
`foo(1+1) * 2`	0

← Python actually returns 0.2999999999998 but I don't expect anyone to know that...

Problem 4: Short Script

Complete the Python script below according to its specification.

```
# analyze_word.py

import p2 # importing module from Problem 2 (you must use dot notation)

"""
When executed, this script prompts the user for a word with at least 5
characters but at most 10 characters long. The script should re-prompt the user
until the user enters such a word.

Once a valid word has been entered, if the word has more vowels than not,
this script prints "Hooray!" and exits. If the word does not have more
vowels than other characters, this script prints "Boo!" and starts
the whole process over again.

Use the function more_vowels that you defined in p2 (see comment above).
"""
done = False
while not done:
    word = input("Enter a word between 5 and 10 letters long: ")
    #check length of word & reprompt if necessary
    while not (5 <= len(word) and len(word) <= 10):
        word = input("Invalid input. Try again: ")
    if p2.more_vowels(word):
        print('Hooray!')
        done = True
    else:
        print('Boo!')
```

There are many solutions. To check yours, try executing your solution on various inputs!