

## Practical Questions

CSCA08 Fall 2017 – Week 3

### Question # 1

Trace through this code using the memory model. What is printed out?

- a) `my_var = 2`  
`print(my_var)`  
`my_var = "Hello"`  
`print(my_var)`
- b) `my_var1 = "twenty"`  
`print(my_var1)`  
`my_var2 = my_var1`  
`my_var1 = 5`  
`print(my_var2)`
- c) `my_var1 = "Brian"`  
`my_var2 = "Rocks"`  
`print(my_var1, my_var2)`  
`my_var3 = my_var1 + " " + my_var2`  
`print(my_var3)`
- d) `def func1 (my_var):`  
`print(my_var)`  
`return my_var`  
  
`my_var = "CSCA08"`  
`print(my_var)`  
`my_var = func1(my_var)`  
`print(my_var)`
- e) `def func1 (my_var):`  
`print(my_var)`  
`my_var = "a"`  
`return my_var`  
  
`my_var = "Car"`  
`print(my_var)`  
`my_var = func1(my_var)`  
`print(my_var)`
- f) `def func1 (my_var):`  
`my_var = my_var *2`  
`return my_var`

```
my_var = "Brian is Awesome "  
print(my_var)  
my_var = func1(my_var)  
print(my_var)
```

g) `def func1 (my_var1, my_var2):`  
    `print(my_var1, my_var2)`  
    `my_var1 = my_var2 *2`  
    `my_var2 = my_var1`  
    `return (my_var1, my_var2)`

```
my_var1 = "A"  
my_var2 = "B"  
print(my_var1, my_var2)  
(my_var2, my_var1) = func1(my_var2, my_var1)  
print(my_var1, my_var2)
```

h) `def func1 (my_var1, my_var2):`  
    `print(my_var1,my_var2)`  
    `my_var3 = func2(my_var1)`  
    `my_var1 = my_var2 *2`  
    `return (my_var1, my_var3)`

```
def func2 (my_var):  
    print(my_var)  
    my_var = "C"  
    my_var = my_var*2  
    return my_var
```

```
my_var1 = "A"  
my_var2 = 4  
print(my_var1, my_var2)  
(my_var1, my_var2) = func1(my_var1, my_var2)  
print(my_var1, my_var2)
```

**i) (Challenge)**

```
def func1 (my_var1, my_var2):  
    print(my_var1,my_var2)  
    my_var1 =func3(my_var1)  
    (my_var2, my_var1) = func2(my_var2, my_var1)  
    my_var2 = func3(my_var2)  
    return (my_var2, my_var1)
```

```
def func2 (my_var1, my_var2):  
    print(my_var2, my_var1)  
    my_var = " B"  
    my_var1 = my_var2  
    my_var2 = my_var + "X"  
    print(my_var1, my_var2)  
    my_var1 = func3(my_var1)  
    return (my_var1,my_var2)
```

```
def func3 (my_var):  
    print(my_var)  
    my_var += " ZYX"  
    my_var = my_var + " ABC"  
    return my_var  
my_var1 = "MC"  
my_var2 = "AD"  
print(my_var1, my_var2)  
(my_var1, my_var2) = func1(my_var1, func3(my_var2))  
print(my_var1, my_var2)
```

```
j) def func1(my_var):  
    my_var += "A"  
    return(my_var)
```

```
def func2(my_var):  
    my_var += "B"  
    return(my_var)
```

```
def func3(my_var):  
    my_var += "C"  
    return(my_var)
```

```
def func4(my_var):  
    my_var += "D"  
    return(my_var)
```

```
def func5(my_var):  
    my_var += "E"  
    return(my_var)
```

```
def func6(my_var):  
    my_var += "F"
```

```
return(my_var)
```

```
def func7(my_var):  
    my_var += "G"  
    return(my_var)
```

```
def func8(my_var):  
    my_var += "H"  
    return(my_var)
```

```
def func9(my_var):  
    my_var += "I"  
    return(my_var)
```

```
def func10(my_var):  
    my_var += "J"  
    return(my_var)
```

```
def func11(my_var):  
    my_var += "K"  
    return(my_var)
```

```
def func12(my_var):  
    my_var += "L"  
    return(my_var)
```

```
def func13(my_var):  
    my_var += "M"  
    return(my_var)
```

```
def func14(my_var):  
    my_var += "N"  
    return(my_var)
```

```
def func15(my_var):  
    my_var += "O"  
    return(my_var)
```

```
def func16(my_var):  
    my_var += "P"  
    return(my_var)
```

```
def func17(my_var):  
    my_var += "Q"  
    return(my_var)
```

```
def func18(my_var):  
    my_var += "R"  
    return(my_var)
```

```
def func19(my_var):  
    my_var += "S"  
    return(my_var)
```

```
def func20(my_var):  
    my_var += "T"  
    return(my_var)
```

```
def func21(my_var):  
    my_var += "U"  
    return(my_var)
```

```
def func22(my_var):  
    my_var += "V"  
    return(my_var)
```

```
def func23(my_var):  
    my_var += "W"  
    return(my_var)
```

```
def func24(my_var):  
    my_var += "X"  
    return(my_var)
```

```
def func25(my_var):  
    my_var += "Y"  
    return(my_var)
```

```
def func26(my_var):  
    my_var += "Z"  
    return(my_var)
```

```
my_var1 = "S"  
my_var2 = "K"  
my_var3 = "U"
```

```
my_var1 = func20(func18(func1(func20(my_var1)))) + “ “
my_var1 =
func19(func20(func14(func5(func13(func14(func7(func9(func19(func19(func1(
my_var1)))))))))) + “ “
my_var1 = func25(func12(func18(func1(func5(my_var1)))))) + “ “
```

```
my_var2 = func16(func5(func5(my_var2))) + “ “
my_var2 = func14(func15(my_var2)) + “ “
my_var2 = func16(func15(func20(my_var2))) + “ “
my_var2 = func6(func15(my_var2)) + “ “
my_var2 = func18(func21(func15(func25(my_var2)))) + “ “
my_var2 = func11(func18(func15(func23(my_var2))))
```

```
my_var3 = func5(func19(my_var3)) + “ “
my_var3 =
func18(func5(func19(func15(func21(func18(func3(func5(func19(my_var3))))))
))) + “ “
my_var3 = func16(func5(func8(func23(my_var3)))) + “ “
my_var3 = func21(func15(func21(my_var3))) + “ “
my_var3 = func20(func5(func7(my_var3))) + “ “
my_var3 = func11(func3(func21(func20(func19(my_var3)))))
```

## Question # 2

Solving for carrots: (<https://open.kattis.com/problems/carrots>)

(You may need some practice getting used to the open kattis rules, but basically you can read using `input()` to get the data they're providing you, and you have to use `print()` to get the answer back to them. You may also find the function `split()` to be helpful (try typing `help(split())` in the shell window)

## Discussion Questions

### Question # 3

Python is a dynamically typed language, which means you don't have to assign a specific type to a variable. In Python you can do assign different data types to one variable:

```
x = "Hello"
```

```
x = 30
```

```
x = True
```

```
x = 3.26
```

```
x = [1, 2, 3, 4, 5]
```

How does this compare to a language, like Java, that requires variables to be of a user-defined types and can't be used for another type without being reassigned?

```
int x = 5
```

```
String y = "hello"
```

What does this mean about the granularity of Python versus a statically-typed language like Java? What are some benefits and drawbacks of a dynamically typed language and a statically-typed language?

#### Question # 4

Give an example of a computer program being stupid (meaning it fails to understand an instruction that seems to be obvious to a human). How would you change this command so that a computer would be able to understand it? What does this tell you about computers?

#### Question # 5

In a statically typed language like Java, you can have 2 functions that have the same name but take 2 different types of data types as parameters (one takes a string and an integer while the other takes a float and an integer). Would you be able to do a similar thing in Python, creating 2 functions with the same name but that took 2 different types of parameters? How would it work?

Java functions:

```
str_num_integrator(int a, String b)
```

```
str_num_integrator(float a, String b)
```

## Logic Questions

Question # 6

Bishops (<https://open.kattis.com/problems/bishops>)

This one is almost purely logical, almost no coding required

Question # 7 (**Challenge**)

*You are sailing the seas and come across a very remote island on your travels. Upon arriving at the island, you are greeted by the inhabitants of the island, 500 majestic unicorn, silver like the moon, each having the same beautiful red eyes, moving and glowing like a forest fire. They are all friendly and are very excited about you visiting as they haven't seen a human in many centuries. They show you around their island and tell you about the unicorn way of life (because of course unicorns can talk).*

*They all seem to be quite normal, normal for unicorns, but then you come across something rather odd. There is a rule on the island which declares that if any unicorn were to ever find out that his/her eyes are red, then at precisely midnight that night, he/she must relinquish all their unicorn powers, lose their horn and special eye color and will be transported into off the island to live their lives in human society as normal horses (which is why we don't see unicorns in our society). However, there are no mirrors on the unicorn's island and the unicorn's never seen to talk about eye color, so they have all been living in blissful ignorance throughout the ages.*

*After experiencing the island, you decide it is time to leave. Upon leaving the island, all the unicorns get together to see you off, and in a tearful farewell you thank them for being such hospitable hosts. Then you decide to tell them*



*something that they all already know (for each can see the colors of the eyes of the other unicorns). Without thinking of any possible consequences, you announce loudly so all the unicorns could hear “at least one of you unicorns has red eyes”. Then you leave, not thinking of possible repercussions (if any). Assuming that the unicorns are (of course) infallibly logical, what happens?*

*If something interesting does happen, what exactly is the new information that you gave the unicorns?*

To get the challenge points of this question, you must post the answer, and an explanation on the Piazza forum. Points will be awarded for the first correct answer, as well as the best explanation(s)

**Note:** This is not a trick question. There's no guessing or lying or discussion by or between unicorns. The answer does not involve any form of sign language, or Mendelian genetics. The answer is logical, and the unicorns are perfectly logical beings. And no, the answer is not "no unicorn transforms."