CIS 110 Optional Review Exercises Answers

Date: 29 Sep 2024
Topic: Functions

Exercise 0 - Conceptual Review

- Functions:
  - What are return types?
  - What is a signature?
    - Ex. def name(x, y):
  - Difference between returning and printing

Exercise 1 - Function headers

Write function headers for the following functions:

1. A function, `printGreeting`, that takes in a name and prints out "Hello [name]!"

```
// Students are not required to write out the function code, but can if
they want to

def printGreeting(name) :
      System.out.println("Hello " + name + "!")
```

2. A function, `outputClass`, that has no arguments and just returns "CIS 110"

```
def outputClass() :
      return "CIS 110"
```

3. A function, `isMultiple`, that takes in two integers and returns true if one is a multiple of the other and false otherwise.

```
def isMultiple(a, b) :
Return (a % b == 0) || (b % a == 0)
```

Exercise 2 - Fill in the blank

An integer is called a SUPER-INDEXED integer if all the integers after the index sum up to that number.

For example:

{1, 3, 2, 1}:

1 is not a super-indexed integer because 3 + 2 + 1 =/= 1

3 is a super-indexed integer because 2 + 1 = 3

2 is not a super-indexed integer because 1 =/= 1

1 is not a super integer because there are no numbers left in the list

The following function takes in an integer list and returns true if there is a single super integer within the list and false otherwise. Fill in the blanks to complete the function

```
def hasSuperIndexedInteger(my_list) :
      for i in range(len(my_list)):
          sub_lst = my_list[i + 1::]
          if sum(sub_lst) == my_list[i]:
             return True

return False
```

Exercise 3 - Tracing

Consider the following function

```
def func(a, b) :
      c = b
      d = 0
      if (b < 0):
            c *= -1

      while (c > 0) :
            d += a
            c-= 1
      if (b < 0) :
            d *= -1
      return d
```

Part A.

What is the output of the following function calls?

1. `func(5, 4)` 20
2. `func(-10, 3)` -30
3. `func(-4, -6)` 24
4. `func(3, 0)` 0

Part B.

Rewrite func using only a single line of code (hint: try to reason about what the function is doing).

```
def func(a, b) :
      return a * b
```

Exercise 4 - Pass by Value

What gets printed out?

```
def func(a, b) :
      a[0] = 24
      b += 1


def main() :
      my_list = {1, 2, 3, 4, 5}
      b = 5
      func(my_list, b)
      print(my_list[0])
      print(b)


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

24
5

For immutables, when passed into a function, they are taken in as parameters. The *literal value* is copied, not the reference.

For sequences like lists, the reference will be passed, but the contents of the list are not copied. Since the reference is passed, any changes to the contents of the list in the function will reflect in the original list.

Exercise 5 - Reverse Function

Write a function `reverse_words` that takes a string sentence as input and returns a new string where every word in the sentence is reversed, but the order of the words remains the same.

```
def reverse_word(sentence):
      #TODO

print(reverse_words("hello world")) # Output: "olleh dlrow"
print(reverse_words("python is fun")) # Output: "nohtyp si nuf"
print(reverse_words("coding is great")) # Output: "gnidoc si taerg"
```

```
def reverse_word(sentence):

    words = sentence.split()
    reverse_words = [word[::-1] for word in words ]
    return “ “.join(reverse_words)
```