

## Operations

1. Assume the following variable declarations have been made, what are the values of each of the following lines of code?

i = 3

j = 10

k = "3"

x = 3.0

y = 2.5

modulus

↳ remainder

$j // i = 3$

\*as string\*

$$\begin{array}{r} 1 \\ \hline 333 \\ \hline 3.333 \\ \hline 3.0 \rightarrow b/c \\ \hline 7.5 \end{array}$$

x is a float

if one of these is a float, the answer is a float

2. Circle any of the following expressions which evaluates to true:

a = 11

b = 30

c = "p"

d = "z"

e = "11"

Things that are false

0

0.0

"" ← empty string

[] ← empty list

11 and "11"

are not the same

i. c > d or a > b

("p" > "z") or (11 > 30) = F

ii. d → True

iii. not (a == b) → True

iv. d != c and a <= b → ("z" != "p") and (11 <= 30) = T

v. a != e → True!

3. What prints out after the following lines are executed?

vowels = ["a", "e", "i", "o", "u"]

i. print(vowels[2:]) through the end

ii. print(vowels[1:4]) up to but not including

iii. print(vowels[:])

["i", "o", "u"]

["e", "i", "o"]

["a", "e", "i", "o", "u"]

vowels[:]

=

vowels[0:]

=

vowels[:len(vowels)]

start at the beginning through the end

4. Which of the following are good ways to name a function? Circle all that apply.

~~i.~~ multnums() → multNums() → camel case!

ii. add()

~~iii.~~ x() → what does this do??

iv. dividingInts()

~~v.~~ FirstMinusSecond()

↑  
FirstMinusSecond()

Code Problem 1:

Read in a file called numbers.txt. This file contains a list of numbers (unsorted), each number on a new line. Find the range and average number without using a list.

General idea:

$$\text{range} = \text{high} - \text{low}$$

$$\text{average} = \frac{\text{sum}}{\text{count}}$$

highNum = -10000

lowNum = 10000

sum = 0

count = 0

FileIn = open("numbers.txt", "r")

for num in FileIn:

num = int(num)

count += 1

sum += num

if num > highNum:

highNum = num

if num < lowNum:

lowNum = num

FileIn.close()

range = highNum - lowNum

average = sum / count

File Reading

1. Open

2. read

3. close

File Writing

1. Open

2. write

3. close

when you read in/take input the type is ALWAYS a string

if statements are independent of each other

if/elifs are mutually exclusive

(only 1 can be true)

↑  
or 0

don't need to strip() since casting to int but wouldn't hurt

\* return is the statement to execute in a function \*

## Problem 2

Write a program that simulates rolling two dice. Ask the user if they would like to specify the number of sides the dice have. If they don't want to, assume they want to roll standard 6-sided dice. After rolling the dice, sum them, and write the number to a file called 'results.txt.'

General Idea:

1. Prompt user  
- if yes, get sides
2. Call `roll()` with number of sides
3. Sum results, write out

```
import random

def roll(numSides=6):
    # returns num rolled
    return random.randrange(numSides) + 1

def main():
    sides = 6
    specify = input("Would you like to specify numSides?")
    if specify.lower() == "y":
        sides = int(input("How many?"))

    sum = 0
    for i in range(2):
        rollOne = roll(sides)
        rollTwo = roll(sides)
        sum = rollOne + rollTwo

    fileOut = open("results.txt", "w")
    print(sum, file=fileOut)
    fileOut.close()
```

no, you don't need to do this on exam

alternatively (1, numSides+1)

using loop

only differences with writing

### Problem 3

What is the difference between a while loop and a for loop? Why would you want to use a for loop as opposed to a while loop? Why would you want to use a while loop as opposed to a for loop? Give an example of the a common 'problem' you'd solve with each loop.

#### For

- when you know how many times you want to iterate
- iterating through sequences
  - strings
  - lists
  - range

word = "hello"

Two main types:  
"Normal":

for letter in word:

letter would be  
h, e, l, l, o

Ranged-based:

for i in range(len(word)):

i would be

0, 1, 2, 3, 4

#### While

- when you don't know how many times you want to loop to repeat

ex.

taking user input

- iterate until the condition is met, however long that might take

#### Problem 4

There are errors in the code below. Try to find them all

1).

while True:

print("Hello")

← NO way to break  
infinite loop

2).

for index in list:

print(index)

list is  
reserved  
word

3).

if myString = "this":

myString = "that"

==

= → assignment operator  
== → comparison

4).

x = random.range(1, 11, 0)

print("I have " + x + " apples")

→ or any positive  
integer

str(x)

concatenating strings  
- left and right side  
must be strings

Single Line Comment symbol: `#comment`

Multi Comments:

```
// // //
comment
// // //
```