Lab 9

Z. Hutchinson
zachary.s.hutchinson@maine.edu

November 12, 2023

The goal of this lab is to get you to think about the problems associated with programming in different ways.

Task 1

Code Haiku. Write a program that is exactly 3 lines long and does something "complex". Complex is yours to define.

Task 2

Give the contents of list3 (in the correct order) after the completion of the following code.

```
def FuncA(A, B):
    C = []
    for a in A:
        for i in range(len(B)-1, -1, -1):
            C.append(a+B[i])
    return C

list1 = [1,2,3]
list2 = [10,100]
list3 = FunA(list1, list2)
```

Task 3

Rewrite the following code so that it does the same thing but is "neater", "clearer" and "easier to read".

```
def Parse(line):
    data = []
    start = 0
    end = 0
    while end < len(line):
        if line[end] == ":":
            if line[start:end]:
                data.append(line[start:end])
            end += 1
            start = end
        else:
            end += 1

    if end > start:
        data.append(line[start:])
    return data
```

Task 4

For each snippet of code, give what is printed. Snippets do not overlap. Consider each snippet a separate program. Give the answer without putting the code through the Python interpreter. Once you have an answer, if you are unsure of it, try the code using the interpreter.

```
# Snippet 1
print( (5 or 1) + (1 and 2) )

# Snippet 2
L = [1]
for i in range(1,5):
    L = L*i
print(len(L))

# Snippet 3
A = [1,2,3,4,5]
B = A[1:3] + A[1:-2]
print(sum(B))

# Snippet 4
c = 0
while c < 10:
    if c % 2 == 0:
        c += 5
    else:
        c -= 3
    print(c)

# Snippet 5
line = "e133 a456 b003 d999 c887"
line = line.split()
m = 0
for i in range(len(line)):
    if int(line[i][1:]) < int(line[m][1:]):
        m = i
```

```
    print(line[m])
```

Task 5

Write a program that will print out the first 10 rows of Pascal's triangle.

https://en.wikipedia.org/wiki/Pascal's_triangle

Task 6

Write a function called *Last10* that has one parameter, *filename*. The parameter *filename* is the name of a file. The function should open the file and using try-except catch the FileNotFoundError. If the file is not found, the function should do nothing. If the file is found, the function should print the last 10 lines of the file. If the file has less than 10 lines, it should print the whole file.