Critical Thinking Corrections and Reflections

Edward Galligan III

ITS320 – Basic Programming

Colorado State University – Global Campus

Dr. Anwar Ahmad

May 2, 2020

Critical Thinking Corrections and Reflections

In every critical thinking assignment, mistakes can be mistakes can be made. To learn from these mistakes, corrections should be made to them to them. Each critical thinking assignment also has a purpose. Lessons learned can be reflected through these learning. Taking these corrections and reflections of lessons learned are important to retain what was learning while doing these assignments.

**Module 1, Option 1: Write a Python Program that Outputs a Mouse**

The first critical thinking assignment involved creating a Python application that had the output of a specific mouse. Although it took a long time to figure out exactly how to do it in Python IDLE and Komodo, this assignment was turned in for full credit. The following is the program from the assignment.

print(' (\-.');\

print(' / \_`>');\

print(' \_) / \_)=');\

print('( / \_/');\

print(' `-.\_\_ (\_\_\_)\_')

The main function learned in this critical assignment was the print() function. The print() function prints text that is inside of quotes inside of the parentheses in print() (Miller, 2016). The text that is inside of the quotes in print() is known as a string literal (Miller, 2016). As an example, print(‘Hello there.’) would have an output of Hello there.

**Module 2, Option 1: Creating a Python Application**

The second critical thinking assignment involved creating a Python program that tells the user if the user inputted string value is true or false for various things; including if the inputted string value contains any alphabetic characters, digits, or lowercase characters; if the user inputted information is less than 50 characters. The following is a corrected critical thinking assignment.

S = str(input('Enter String Value: '))

if len(S) > 50:

print('Error! String cannot be longer than 50 characters.')

if len (S) < 50:

print('Alphanumeric?: %s'%S.isalnum())

print('Alphabetical?: %s'%S.isalpha())

print('Digits?: %s'%S.isdigit())

print('Lower Case?: %s'%S.islower())

print('Upper Case?: %s'%S.isupper())

The main functions learned in this critical assignment is the use of branching, relational operators, and string methods. The branching used in the program are if expressions. Branching directs a Python program to execute statements if the expression is true or not (Miller, 2016). The relational operator used in this program is < (less-than). Relational operators are evaluated on a Boolean value, which means the statement is either True or False (Miller, 2016). This string methods this program uses check the string for alphanumeric, alphabetical, digit, lowercase, and uppercase characters; represented by isalnum, isalpha, isdigit, islower, and isupper; respectively (Python, n.d.)

**Module 3, Option 1: Creating a Program to Calculate the Value of a Ferrari**

The third critical thinking assignment involved implementing a program that takes an inputted year by the user and outputs an approximate value of a Ferrari from the given year. This assignment was turned in for full credit. The following is the program from the assignment.

year = int(input('Enter year:\n'))

if year < 1962:

print('Car did not exist yet!')

elif year <= 1964:

print('$', 18500)

elif year <= 1968:

print('$', 6000)

elif year <= 1971:

print('$', 12000)

elif year <= 1975:

print('$', 48000)

elif year <= 1980:

print('$', 200000)

elif year <= 1985:

print('$', 650000)

elif year <= 2012:

print('$', 35000000)

elif year <= 2014:

print('$', 52000000)

The main functions learned in this critical thinking assignment were the use of relational operators and the use of multiple elifs. The two relational operators used in this critical thinking assignment are < and <=, which are described as less-than and less-than-or-equal-to, respectively (Miller, 2016). Relational operators involve two operands which are evaluated to a Boolean value, meaning that the operator is either true or false (Miller, 2016).When a programmer requires more than two branches, an elif is used, which is short for else if (Miller, 2016). Any number of elif terms can be used (Miller, 2016)

**Module 4, Option 1: Repetition Control Structure – Five Floating Point Numbers**

The fourth critical thinking assignment involved writing a program that utilized a loop that read a set of five floating-point values that were inputted from the user. The program would then take these outputs to show the total of the values, average of the values, maximum value, minimum value, and a 20% increase to each value. This assignment was turned in for full credit. The following is the program from the assignment.

amounts = int(input('Enter five floating-point values:\n'))

for number in range(0, len(amounts)):

total += amounts(number)

print('The total is', total)

average = total / len(amounts)

print('The average is', average)

maximum = max(amounts)

print('The highest value is', maximum)

minimum = min(amounts)

print('The lowest value is', minimum)

for number in amounts:

amounts\_with\_interest.append(number \* 1.2)

print('Original values with 20% interest are', amounts\_with\_interest)

The main functios learned in this critical thinking assignment were the use of for loops and range(). For loops loop over each element in a container, one at a time, assigning the next element to a variable that can then be used in the loop’s body (Miller, 2016). The container for the for loop statement is usually a list, tuple, or string (Miller, 2016). Using the range() function allows for counting in for loops (Miller, 2016).

**Module 5, Option 1: String Values in Reverse Order**

The fifth critical thinking assignment involved writing a program that takes three user inputted string values and prints the string values in reverse order. This assignment was turned in for full credit. The following is the program from the assignment.

user\_str = str(input('Enter three string values:\n'))

def main\_method(user\_str):

return reversed(user\_str)

print('String values in reverse order is:', main\_method)

The main functions learned in this critical thinking assignment were using the def, return, and reversed functions. The def keywork allows a programmer to create a new user-defined function (Miller, 2016). A function’s return causes execution to jump to the function’s statement (Miller, 2016). By using reversed(), a program will reverse the order of elements in a list to iterate the elements backwards (Miller, 2016).

**Module 6, Option 1: Working with Python Classes**

The sixth critical thinking assignment involved writing a program that takes two complex numbers and prints the results of their addition, subtraction, multiplication, division, and modulus operations. The following is a corrected critical thinking assignment.

class nums(object):

def \_\_init\_\_(self, real, imaginary):

self.real = real

self.imaginary = imaginary

def add(self):

return self.real + self.imaginary

def sub(self):

return self.real - self.imaginary

def mul(self):

return self.real \* self.imaginary

def div(self):

return self.real / self.imaginary

def mod(self):

return self.real % self.imaginary

if \_\_name\_\_ == '\_\_main\_\_':

#Input two numbers

var1, var2 = input("Enter two numbers giving a space here: ").split()

#Convert to floats

var1, var2 = float(var1), float(var2)

#Create Instance

a = nums(var1, var2)

#Print inputs

print('Inputs: ', a.real, a.imaginary)

#Print Sum

print('Sum: ', a.real, '+', a.imaginary, 'i', '='

, a.add())

#Print Sub

print('Sub: ', a.real, '-', a.imaginary, 'i', '='

, a.sub())

# Print Mul

print('Mul: ', a.real, '\*', a.imaginary, 'i', '='

, a.mul())

# Print Div

print('Div: ', a.real, '/', a.imaginary, 'i', '='

, a.div())

# Print Mod

print('Mod: ', a.real, '%', a.imaginary, 'i', '='

, a.mod())

The main function learned in this critical thinking assignment was using classes. The class keyword is used in Python to create a user-defined type of object that contains groups of related variables and functions (Miller, 2016). A created class creates a new type of object that maintains a set of attributes that determine the data and behavior from the class (Miller, 2016).

**Conclusion**

By making corrections to programs, reflecting over purposes of assignments, and reiterating functions learned when working on programs, a programmer learning to use Python can ensure mastery of writing programs in Python.

References

Miller, B. (2016). *Programming in Python 3 with ZyLabs*. https://csuglobal.instructure.com/courses/20287/external\_tools/28168

Python. (n.d.). *Python Documentation*. https://docs.python.org/2/library/stdtypes.html#string-methods