PYTHON ASSIGNMENT BOOK

MAKE A MOVE TO PYTHON

**ASSIGNMENTS**

**TASK ONE: NUMBERS AND VAR****IABLES**

1. Create three variables in a single line and assign different values to them and make sure their data types are invited different. Like one is int, another one is float and last one is string.

- a, b , c = “A”, 1, 1.5

2. Create a variable of value type complex and swap it with another variable whose value is an integer.

a = 1+2x

b = 2

a=b

3. Swap two numbers using third variable as result name and do the same task without using any third variable.

1st: a,b = 1,2

result = a

a = b

b = result

2nd:

a,b = 2,3

a = a+b # a = 5

b = a-b # b = 2

a = a -b # a = 2

print("a and b = ",str(a),"and",str(b))

4. Write a program to print the value given by the user by using both Python 2.x and Python 3.x Version.

def Python\_2(string):

Print string

def Python\_3(string):

Print (string)

Python3(“Hello”)

Python2(“Hello”)

5. Write a program to complete the task given below:

* Ask user to enter any 2 numbers in between 1-10 and add both of them to another variable call z.

def UserInput(a,b):

z = a+b

return(z)

UserInput(2,4)

* Use z for adding 30 into it and print the final result by using variable result.

def UserInput(a,b):

z = a+b

z = z+30

return(z)

UserInput(2,4)

6. Write a program to check the data type of the entered values. HINT: Printed output should say - The input value data type is : int/float/string/etc

- a = 10

type(a)

7. Create Variable using CamelCase, LadderCase and UPPERCASE. (Refer: <https://capitalizemytitle.com/camel-case/>)

Name = “Payaj”

CamelCase: Name.title()

UpperCase: Name.upper()

LadderCase: Name.lower() # don’t know what a ladder case is, guessing it’s lower.

8. If one data type value is assigned to ‘a’ variable and then a different data type value is assigned to ‘a’ again. Will it change the value. If Yes then Why?

Yes it will change the value of the variable because you can overwrite the variables.

**TASK TWO: OPERATORS AND DECISION MAKING STATEMENT**

1. Write a program in Python to perform the following operation:

* If a number is divisible by 3 it should print “Consultadd” as a string

def number(a):

if(a %3 == 0):

return(“Consultadd”)

print(number(9))

* If a number is divisible by 5 it should print “c” as a string

def number(a):

if(a %5 == 0):

return(“c”)

print(number(10))

* If a number is divisible by both 3 and 5 it should print “Consultadd Python Training” as a string.

def number(a):

if(a %5 == 0 and a % 3 == 0):

return(“Consultadd”)

print(number(10))

2. Write a program in Python to perform the following operator based task:

* Ask user to choose the following option first:
  + If User Enter 1 - Addition
  + If User Enter 2 - Subtraction
  + If User Enter 3 - Division
  + If USer Enter 4 - Multiplication
  + If User Enter 5 – Average

def userInput(first,second,num):

value = “”

if(num == 1):

value = first+second

operation = “Addition”

elif(num == 2):

value = first-second

operation =Subtraction”

elif(num == 3):

value = first/second

operation = “Division”

elif(num == 4):

value = first\*second

operation = “Multiplication”

else:

value = (first+second)/2

operation = “Average”

if(value >= 0):

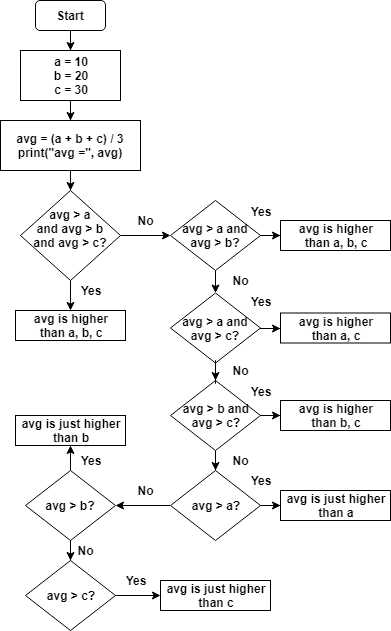
return (operation,“ = ”, str(value))

if(value < 0):

return (operation,“ = NEGATIVE”)

* Ask user to enter the 2 numbers in a variable for first and second for the first 4 options mentioned above.
  + print(userInput(first=3,second=2,num=1))
  + print(userInput(first=3,second =2,num=2))
  + print(userInput(first=3,second =2,num=3))
  + print(userInput(first=3,second =2,num=4))
* Ask user to enter two more numbers as first and second2 for calculating the average as soon as user choose an option 5.
  + print(userInput(first=3,second =2,num=5))
* At the end if the answer of any operation is Negative print a statement saying “NEGATIVE”
  + print(userInput(first=2,second =3,num=2))
* NOTE: At a time user can perform one action at a time.

3. Write a program in Python to implement the given flowchart:



a, b, c = 10,20,30

avg = (a+b+c)/3

print(“avg = “+str(avg))

if((avg>a and avg > b and avg > c)):

print (“avg is higher than a, b, c”)

else:

if(avg>a and avg>b):

print (“avg is higher than a, b, c”) # are you sure it’s “a,b,c” and not “a,b” in the flow chart)

elif(avg > a and avg > c):

print((“avg is higher than a, c”)

elif(avg > b and avg > c):

print((“avg is higher than b, c”)

elif(avg > a):

print((“avg is higher than a”)

elif(avg > b):

print((“avg is higher than b”)

else:

print((“avg is higher than c”)

4. Write a program in Python to break and continue if the following cases occurs:

* If user enters a negative number just break the loop and print “It’s Over”
  + for value in [1,2,3,-1,5]:

if (value <0):

print(“it’s Over”)

break

* If user enters a positive number just continue in the loop and print “Good Going”
  + for value in [1,2,3,5]:

if (value > 0):

print(“Good Going”)

continue

5. Write a program in Python which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200.

For num in range(2000,3201):

If ((num%7 == 0) and (num%5 != 0) ):

Print(num)

6. What is the output of the following code examples?

* x=123

for i in x:

print(i)

* It’ll give an error as 123 is an integer, which is not iterable.
* i = 0

while i < 5:

print(i)

i += 1

if i == 3:

break

else:

print(“error”)

result: 0 1 2

* count = 0

while True:

print(count)

count += 1

if count >= 5:

Break

Result : 0,1,2,3,4

7. Write a program that prints all the numbers from 0 to 6 except 3 and 6.

Expected output: 0 1 2 4 5

Note: Use ‘continue’ statement

for num in range(7):

if num % 3 == 0:

print(num)

continue

8. Write a program that accepts a string as an input from user and calculate the number of digits and letters.

Expected output: consul12

Letters 6

Digits 2

Letters = 0

Digits = 0

For letter in “consul12”:

If letter.isdigit():

Digits += 1

If letter.isalpha():

Letter += 1

Print(“Letters”,Letters)

Print(“Digits”,Digits)

9. Read the two parts of the question below:

* Write a program such that it asks users to “guess the lucky number”. If the correct number is guessed the program stops, otherwise it continues forever.

def guessNum(number):

While True:

if number == 6:

return(“Bingo”)

* Modify the program so that it asks users whether they want to guess again each time. Use two variables, ‘number’ for the number and ‘answer’ for the answer to the question whether they want to continue guessing. The program stops if the user guesses the correct number or answers “no”. ( The program continues as long as a user has not answered “no” and has not guessed the correct number)

def guessNum(number,answer):

print(answer)

if number == 6:

return("Bingo")

elif answer == "no":

return ("Game Over")

else:

return("Do you want to guess again?")

number = 5

answer = "no"

guessNum(number,answer)

10. Write a program that asks five times to guess the lucky number. Use a while loop and a counter, such as

counter=1

While counter <= 5:

print(“Type in the”, counter, “number”

counter=counter+1

The program asks for five guesses (no matter whether the correct number was guessed or not). If the correct number is guessed, the program outputs “Good guess!”, otherwise it outputs “Try again!”. After the fifth guess it stops and prints “Game over!”.

count = ""

def guessNum(number,answer, count):

if count <= 5:

if number == 6:

return(["Goog Guess!",count])

elif answer == "no":

return (["Game Over!",count])

else:

if(count == 5):

count += 1

return(["Sorry but that was not very successful"])

else:

print("Attempt number : ", str(count))

count += 1

return(["Do you want to guess again?",count])

else:

return(["No attempt left",count])

number = 5

answer = "yes"

if(count == ""):

Enter\_Value = guessNum(number,answer, 1)

count = Enter\_Value[1]

print(Enter\_Value[0])

else:

Enter\_Value = guessNum(number, answer, count)

count = Enter\_Value[1]

print(Enter\_Value[0])

11. In the previous question, insert “break” after the “Good guess!” print statement. “break” will terminate the while loop so that users do not have to continue guessing after they found the number. If the user does not guess the number at all, print “Sorry but that was not very successful”.

count = ""

def guessNum(number,answer, count):

if count <= 5:

if number == 6:

return(["Goog Guess!",count])

elif answer == "no":

return (["Game Over!",count])

else:

if(count == 5):

count += 1

return(["Sorry but that was not very successful"])

else:

print("Attempt number : ", str(count))

count += 1

return(["Do you want to guess again?",count])

else:

return(["No attempt left",count])

number = 5

answer = "yes"

if(count == ""):

Enter\_Value = guessNum(number,answer, 1)

count = Enter\_Value[1]

print(Enter\_Value[0])

else:

Enter\_Value = guessNum(number, answer, count)

count = Enter\_Value[1]

print(Enter\_Value[0])

**TASK THREE: DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.

- ListSample = [1,2,.3,”a”,1+4j,2.3,”p”,”v”,0,”Letters”]

2. Create a list of size 5 and execute the slicing structure

- listSample = [‘a’,’b’,’c’,’d’,’e’]

Print(listSample[:3])

3. Write a program to get the sum and multiply of all the items in a given list.

Sum([1,2,3,4,5])

P = 1

for num in [1,2,3,4,5]:

p = p\*num

print(p)

4. Find the largest and smallest number from a given list.

max([1,2,3])

min([1,2,3])

5. Create a new list which contains the specified numbers after removing the even numbers from a predefined list.

- newList = []

for num in [1,2,3,4,5,6,7,8]:

if num%2 != 0:

newList.append(num)

6. Create a list of first and last 5 elements where the values are square of numbers between 1 and30 (both included).

newList = []

for num in range(1,31):

newList.append(num\*\*2)

print(newList)

7. Write a program to replace the last element in a list with another list.

Sample data: [[1,3,5,7,9,10],[2,4,6,8]]

Expected output: [1,3,5,7,9,2,4,6,8]

- SampleData[0][:-1]+ SampleData[1]

8. Create a new dictionary by concatenating the following two dictionaries:

a={1:10,2:20}

b={3:30,4:40}

Expected Result: {1:10,2:20,3:30,4:40}

- a.update(b)

9. Create a dictionary that contains a number (between 1 and n) in the form(x,x\*x).

Sample data (n=5)

Expected Output: {1:1,2:4,3:9,4:16,5:25}

dic = {}

for i in range(1,6):

dic.update({i:i\*\*2})

10. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program:

34,67,55,33,12,98

The output should be:

[‘34’,’67’,’55’,’33’,’12’,’98’]

(‘34’,’67’,’55’,’33’,’12’,’98’)

Ans:

List = “34,67,55,33,12,98”.split(“,”)

Tuple = tuple(List)

**MORE QUESTIONS ON DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.

- newList = [“x”, 1, 1.2, 2+1j, 3-1j, “abc” , “1” , “string”, 2, 4.1]

2. Create a list of size 5 and execute the slicing structure

- newList = [1,2,3,4,5]

Print(newList[2:4])

3. Create a list of given structure and run

**x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]**

* Access list [1, 2, 3, 4]
  + - X[5][:4]
* Access list [600,  700]
  + - X[6:8]
* Access list [100, 300, 500, 600, 800]
  + - x[slice(0,5,2)]+x[slice(6,len(x),2)]
* Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]]
  + - [list(x[::-1])]
* Access list [10]
  + - x[5][5][0:1]
* Access list [ ]
  + - x.clear

Print(x)

4. Create a list of thousand number using range and xrange and see the difference between each other.

- x = range(1,1001)

Y = list(xrange(1001,2001))

[b-a for a,b in zip(x,y)]

5. How Tuple is beneficial as compare to the list?

- Tuple requires less memory because it’s immutable and is also faster than list. Tuple can also be a value in set because of it’s immutable.

6. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and a multiple of 2.

For num in range(1,101):

If num % 3 == 0 and num % 2 == 0:

Print(num)

7. Write a program in Python to reverse a string and print only the vowel alphabet if exist in the string with their index.

String= "String"

for i in range(len(String)):

if String[i] in "aeiou":

print("index = ",str(len(String)-1-i),"value",String[len(String)-1-i])

8. Write a program in Python to iterate through the string “hello my name is abcde” and print the string which has even length of word.

For s in “hello my name is abcd”.split(“ “):

if len(s) %2 :

print(s)

9. Write a program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

**x=[1,2,3,4,5,6,7,8,9,-1]**

for i in range(len(x)):

for j in range(i+1,len(x)):

if x[i]+x[j] == 8:

print("pair that gives sum of 8: ",[x[i],x[j]])

10. Write a program in Python to complete the following task:

* Create two different list as in even\_list and odd\_list

even\_list, odd\_list = [],[]

for i in range(1,10):

if i % 2 == 0:

even\_list.append(i)

else:

odd\_list.append(i)

* Ask user to enter the number in the range of 1,50 and make sure if the entered number is even append it to the even\_list and if the entered number is odd append it to the odd list.

even\_list, odd\_list = [],[]

while True:

i = int(input("Enter a number between 1 and 50"))

if i % 2 == 0 and i >= 1 and i <= 50:

even\_list.append(i)

elif i >= 1 and i <= 50:

odd\_list.append(i)

break

* Keep that in mind you can only add 5 items in each list

even\_list, odd\_list = [],[]

while True:

i = int(input("Enter a number between 1 and 50"))

if i % 2 == 0 and i >= 1 and i <= 50 and len(even\_list)<5:

even\_list.append(i)

print("you only add ",str(5-len(even\_list)), " more even numbers")

if i % 2 != 0 and i >= 1 and i <= 50 and len(odd\_list)<5:

odd\_list.append(i)

print("you only add ", str(5 - len(odd\_list)), " more odd numbers")

if len(even\_list) == 5 and len(odd\_list) ==5:

break

* Make sure once you entered the total 5 element calculate the sum of the list and return the maximum out of the list.

even\_list, odd\_list = [],[]

oddSum, evenSum = 0,0

def evenOddList():

while True:

i = int(input("Enter a number between 1 and 50"))

if i % 2 == 0 and i >= 1 and i <= 50 and len(even\_list)<5:

even\_list.append(i)

print("you only add ",str(5-len(even\_list)), " more even numbers")

if i % 2 != 0 and i >= 1 and i <= 50 and len(odd\_list)<5:

odd\_list.append(i)

print("you only add ", str(5 - len(odd\_list)), " more odd numbers")

if len(even\_list) == 5 and len(odd\_list) ==5:

oddSum = sum(odd\_list)

evenSum = sum(even\_list)

return max(oddSum, evenSum)

print(evenOddList())

11. Write a program to find out the occurrence of a specific word from an alphanumeric statement. **Example:** 12abcbacbaba344ab

**Output:** a=5 b=5 c=2 make sure you should avoid the numbers in you logic

string = "12abcbacbaba344ab"

for i in set(string):

if i.isalpha():

print(i," = ",str(string.count(i)))

12.          Generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

tup = (1,2,3,4,5,6,7,8,9,10)

new\_tup = tuple()

for i in tup:

if i%2 == 0:

new\_tup=new\_tup + (i,)

**TASK FOUR: FUNCTIONS**

1. Write a program to reverse a string.

Sample data: “1234abcd”

Expected Output: “dcba4321”

string = "1234abcd"

def revString(string):

newSting = ""

for i in range(len(string)):

newSting+=string[len(string)-1-i]

return newSting

print(revString("1234abcd"))

2. Write a function that accepts a string and calculate the number of uppercase letters and lowercase letters.

Expected Output:

No. of Upper case characters : 3

No. of Lower case Characters : 12

def stringCase(string):

loweCase = 0

upperCase = 0

for s in string:

if s.islower():

loweCase += 1

elif s.isupper():

upperCase += 1

return [loweCase,upperCase]

print("No. of upper case characters : "+ str(stringCase("SampleString")[1]))

print("No. of lower case characters : "+ str(stringCase("SampleString")[0]))

3.        Create a function that takes a list and returns a new list with unique elements of the first list.

def stringCase(List):

newList = list(set(List))

return newList

print(stringCase(["a","b","a","c"]))

4.         Write a program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

def stringCase(string):

string = string.split("-")

string.sort()

return "-".join(string)

print(stringCase("a-c-d-b"))

5.         Write a program that accepts a sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

Sample input:

Hello world

Practice makes perfect

Expected Output:

HELLO WORLD

PRACTICE MAKES PERFECT

def stringCase(string):

string = string.upper()

return (string)

print(stringCase("Hello world\nPractice makes perfect"))

6.          Define a function that can receive two integral numbers in string form and compute their sum and print it in console.

def Sum(a,b):

add = float(a)+float(b)

return (add)

print(Sum("1","2"))

7.        Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print all strings line by line.

def BiggestString(a,b):

if len(a) > len(b):

return (a)

else:

return b

print(BiggestString("1hvhjb","222"))

8.        Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20.

def Tup():

tup = tuple()

for i in range(1,21):

tup = tup+(i\*\*2,)

return tup

print(Tup())

9.         Write a function called showNumbers that takes a parameter called limit. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers.

Example: If the limit is 3 , it should print:

0 EVEN

1 ODD

2 EVEN

3 ODD

def showNumbers(limit):

for i in range(limit+1):

if i%2 ==0:

print(i,"EVEN")

else:

print(i,"ODD")

return

(showNumbers(3))

10. Write a program which can filter() to make a list whose elements are even number between 1 and 20 ( both included)

a = filter(lambda x:x%2==0,range(1,21))

print(list(a))

11. Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10]

Hints: Use map() to generate a list.

          Use filter() to filter elements of a list

            Use lambda to define anonymous functions

sequence = range(1,11)

result = filter(lambda y:y%2==0,map(lambda x:x\*\*2, sequence))

print(list(result))

12. Write a function to compute 5/0 and use try/except to catch the exceptions

def Infinity(a,b):

try:

return b/a

except Exception as e:

return("Exception: "+str(e))

print(Infinity(0,5))

13. Flatten the list [[1,2,3],[4,5],[6,7,8]] into [1,2,3,4,5,6,7,8] using reduce

Goal : Turn [1,2,3,4,5,6,7] to 1234567

List = [[1,2,3],[4,5],[6,7,8]]

x=[]

for i in List:

x = x+i

String = "".join(map(str,x))

print(int(String))

 14. What is the output of the following codes:

(i) def foo():

    try:

        return 1

    finally:

        return 2

k = foo()

print(k)

Ans: 2

(ii) def a():

    try:

        f(x, 4)

    finally:

        print('after f')

    print('after f?')

a()

Ans: after f

**TASK FIVE: FILE HANDLING AND EXCEPTION HANDLING**

1. Write a program in Python to allow the error of syntax to go in exception. HINT: use SyntaxError

2. Write a program in Python to allow user to open a file by using argv module. If the entered name is incorrect throw an exception and ask them to enter the name again. Make sure to use read only mode.

3. Write a program to handle an error if the user entered the number more than four digits it should return “Please length is too short/long !!! Please provide only four digits”

4. Create a login page backend to ask user to enter the UserEmail and password. Make sure to ask Re-Type Password and if the password is incorrect give chance to enter it again but it should not be more than 3 times.

5. <https://www.programiz.com/python-programming/exception-handling> Go through this link to understand Finally and Raise concept.

6. Read any file using Python File handling concept and return only the even length string from the doc.txt file.  
Consider the content as:

Hello I am a file

Where you need to return the data string

Which is of even length

Make sure you return the content in

The same link as it is present.