4MM013 Python Workshop-1
Feb 25, 2024

Rewrite the following programs (Example 1 to 16) on jupyter Notebook (preferred). Test the code and ensure it runs correctly, and explain what these code do through the comments.

Please write your name and University Id in the first cell of your workshop file.

Let's get started with some nice and easy questions

Variables:

Example 1:

```
length = 10 #variable1

width = 5  #variable2

area = length * width  #This is called an expression

print(area)
```

- ✓ *Variables will have no spaces, No special characters like, $, #. %, ^ etc*
- ✓ *It must start with letter or an underscore(_)*
- ✓ *Name, nAME, name these all are valid variable names (case-sensitive)*

Type conversion:

- ✓ **Implicit Conversion**

  Example 2:

```
print(7+8.5)   # 7 is int type & 8.5 is float type, but mixing these two programs run well
```

- ✓ **Explicit Conversion**

  Example 3:

```
base = 6

height = 2

area = base*height/2

print("Area is " + str(area)) #changed, int type, 'area' to string
```

1. Write a program to prompt the user for **hours** and rate per **hour** using input( ) to compute gross pay. Pay the hourly rate for the hours up to 40 and 1.5 times the hourly rate for all hours worked above 40 hours. Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75).

```
hrs = input("Enter Hours:")
h = float(hrs)
rate = input("Enter the rate per hour:")
r = float(rate)
# complete the program
```

Function:

Example 4:

```
def greeting (name): #passing parameters/argumets to the function
    print("welcome "+ name)
greeting("Type_Your_Name")
```

Example 5:

```
## getting values out of the function(return values)
def area_traingle(b, h):
    return b*h/2
area = area_traingle(4, 8)
print("area of traingle is: " + str(area))
```

2. Write a function that compares two numbers and returns them in increasing order.

Example 6:

```
def order_numbers(number1, number2):
    if number2 > number1:
        return ??
    else:
        return ??
# 1) Fill in the blanks so the print statement displays the
result of the function call
smaller, bigger = order_numbers(100, 99)
print(smaller, bigger)
```

Conditionals:

Example-7

```
#Even or Odd
def is_Even(number):
    if (number%2 == 0):
        return True
    return False
```

```
is_Even(7)
```

Example-8

```
def number_group(number):
    if number > 0:
        return "Positive"
    elif number == 0:
        return "Zero"
    else:
        return "Negative"
print(number_group(10)) #Should be Positive
print(number_group(0)) #Should be Zero
print(number_group(-5)) #Should be Negative
```

Loops:

While:

Example-9

```
#while loop
x = 0  # initialzation of variable  Why=?
while x<5:
    print("Not there yet, x= " + str(x))
    x = x+1
print("x= "+str(x)) #output=??
```

Example-10

```
def count_down(start_number):
current=3 #initialzation of variable
while (current > 0):
     print(current)
     current -= 1
print("Zero!")
```

```
count_down(3) #output=??
```

Note: When you are writing a loop, check that you are initializing all the variables you want to use before you use them.

***Common mistakes**:*

- ✓ *Forgetting to initialize variables. If you try to use a variable without first initializing it, you'll run into a NameError.*
- ✓ forgetting to initialize variables with the correct value could result infinite loop

Example-11

```
#Infinite loops and how to break them?
while x%2 == 0:
    x = x/2
#what happens here when x = 0 (infinite loop)
#can you solve this?
```

Example-12

```
#Solution of Example 11

if x != 0:

    while x%2 == 0:

        x = x/2

#or

while x != 0 and x%2 == 0:

    x = x/2
```

For Loop:

Example-13

```
# for loop, iterates over a sequence of values

for x in range(5):

    print(x) #Output=??
```

Example-14

```
def square(n):

    return n*n

def sum_squares(x):
```

```
    sum = 0
    for n in range(10):
        sum += square(n)
    return sum
print(sum_squares(10)) #Output=??
```

Explain what the program does?

Example-15

```
values = [10, 20, 30, 40, 50]
Sum = 0
length = 0
for value in values:
    Sum = Sum+value
    length = length+1
print("Total sum= "+str(Sum)+"and the Average="+str(Sum/length))
#Output=??
```

Explain what the program does?

Note:

Use for loops: when there is a sequence of elements that you want to iterate
Use while loops: when you want to repeat an action until a condition changes

The range() function when passing one, two, or three parameters:

- ✓ One parameter will create a sequence, one-by-one, from zero to one less than the parameter.
- ✓ Two parameters will create a sequence, one-by-one, from the first parameter to one less than the second parameter.
- ✓ Three parameters will create a sequence starting with the first parameter and stopping before the second parameter, but this time increasing each step by the third parameter.

Example-16

```
#nested for loops
for left in range(7):
    for right in range(left, 7):
        print("["+str(left)+"|"+str(right)+"]", end=" ")
    print()
#Output=?? Explain what the program does?
```

Exercises: 1

a. Write a function fib (n) that prompts the number of terms "n" from the user and displays first "n" terms of Fibonacci sequence iteratively.
b. Write a function *myPow(x, n)* that returns x^n, where and n is a non-negative integer. Do not use the ** operator or the math.pow function - use one *while* loop.
c. Write a function SumDigits(n) that displays the sum of digits of "n". Hint: SumDigits(234) will be 2+3+4=9
d. Write a function *squaredSum* that returns the sum of the squares of an array/list of integers such that *squaredSum([2,3,4])* gives 29. Write your solution in pure Python to show the logic in your solution, do not use any library functions.
e. Write a function called *sumDigits* that is given an integer *num* and returns the sum of the digits of *num*. Eg. *sumDigits(6)* gives 21. Do not use any library functions (if there is one).

"The End"