HW 1: Basic Python Programming

CPE232 Data Models

Owner : 66070501043 - Phoorin Chinphuad

1. Basic usage

> John Doe is a 29 years-old system engineer who earns ฿41500.00 a month.

Create and assign variables to store this person's information (name, age, position and salary).

```python
# Write your code here
name = "John Doe"
age = 29
position = "System Engineer"
salary = 41500.00
```

What is the type of each variables?

```python
# Write your code here
print(f"name type is : {type(name)}")
print(f"age type is : {type(age)}")
print(f"position type is : {type(position)}")
print(f"salary type is : {type(salary)}")
```

```
name type is : <class 'str'>
age type is : <class 'int'>
position type is : <class 'str'>
salary type is : <class 'float'>
```

The manager decides to give John a 7% raise. Update his salary.

```python
# Write your code here
salary = salary * 1.07
```

Prints his information again with his new salary.

```python
# Write your code here
print(f"{name} is a {age} years-old {position} who earns ฿{salary} a month.")
```

```
John Doe is a 29 years-old System Engineer who earns ฿44405.0 a month.
```

Now, he decides to resign. Delete his information from the system.

```python
# Write your code here
del name
del age
del position
del salary
```

2. Variable and Expression

2.1 Write a code to convert temperature unit from celcius to other units

```python
C = 34.5
print(C)
```

```
34.5
```

Fahrenheit

$$\frac{C}{5} = \frac{F-32}{9}$$

```
F = (C * 9/5) + 32
print(F)
```

```
94.1
```

Kelvin
$K = C + 273.15$

```
K = C + 273.15
print(K)
```

```
307.65
```

Rømer

$$Ro = \frac{C \times 21}{40} + 7.5$$

```
Ro = (C * 21/40) + 7.5
print(Ro)
```

```
25.6125
```

3. Multi-item variables

List

```
names = ['Thomas', 'Kate', 'Mike', 'Amelia', 'James', 'Megan']
```

Create new variable call `new_name` which takes input name of the user.

```
new_name = input('Enter your name: ')
```

```
Enter your name: Feen
```

Insert `new_name` into `names` list.

```
# Write your code here
names.append(new_name)
```

Select your name from the list

```
# Write your code here
names[-1]
```

```
'Feen'
```

Merge `another_names` into `names`.

```
another_names = ['Peter', 'Steve', 'Sam', 'Charlotte']
```

```
# Write your code here
names += another_names
print(names)
```

```
['Thomas', 'Kate', 'Mike', 'Amelia', 'James', 'Megan', 'Feen', 'Peter', 'Steve', 'Sam', 'Charlotte']
```

Change `Amelia`'s name to `Amy`

```python
# Write your code here
names[3] = "Amy"
print(names)
```

```
['Thomas', 'Kate', 'Mike', 'Amy', 'James', 'Megan', 'Feen', 'Peter', 'Steve', 'Sam', 'Charlotte']
```

Dictionary

```python
capital_city = {'England':'London',
                'Spain':'Madrid',
                'Japan':'Tokyo',
                'Australia':'Sydney',
                'Germany':'Berlin',
               }
```

Add a record `Thailand` and it's capital city to this dictionary

```python
# Write your code here
capital_city['Thailand'] = 'Bangkok'
print(capital_city)
```

```
{'England': 'London', 'Spain': 'Madrid', 'Japan': 'Tokyo', 'Australia': 'Sydney', 'Germany': 'Berlin', 'Thailand': 'Bangkok'}
```

You may notice that the capital city of `Australia` is wrong. It should be `Canberra`. Correct this mistake.

```python
# Write your code here
capital_city['Australia'] = 'Canberra'
print(capital_city)
```

```
{'England': 'London', 'Spain': 'Madrid', 'Japan': 'Tokyo', 'Australia': 'Canberra', 'Germany': 'Berlin', 'Thailand': 'Bangkok'}
```

4. Control Flows and conditional statements

if...elif...else

1. Define a variable to get input age from user.

```python
age = int(input("Enter your age >> "))
```

```
Enter your age >> 19
```

Write a series of if...elif...else statement that categorize input age into following groups:

> Babies: 0-2 years old
> Children: 3-12 years old
> Teenager: 13-19 years old
> Young Adults: 20-29 years old
> Middle-aged Adults: 30-45 years old
> Old Adult: 46-59 years old
> Elderly: Above 60 years old

```python
# Write your code here
if((age >= 0) & (age <= 2)):
    print("Babies")
elif(age <= 12):
    print("Children")
elif(age <= 19):
    print("Teenager")
elif(age <= 29):
    print("Young Adults")
elif(age <= 45):
    print("Middle-aged Adults")
elif(age <= 59):
    print("Old Adult")
```

```
else:
    print("Elderly")
```

```
Teenager
```

Looping

1. Write a code to create a multiplication table of an input number (multiplier from 1-12).

```
# Write your code here
num = int(input("Enter a number: "))
for i in range(1,13):
    print(f"{num} x {i} = {num*i}")
```

```
Enter a number: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
5 x 11 = 55
5 x 12 = 60
```

2. Write a code that construct the following pattern.

ไม่รองรับประเภทเซลล์ ดับเบิลคลิกเพื่อตรวจสอบ/แก้ไขเนื้อหา

```
# Write your code here
num = int(input("Enter a number: "))
for i in range(1,num+1):
    print("*" * i)
```

```
Enter a number: 5
*
**
***
****
*****
```

3. Creates a loop to print `I love <programming language>!` except for Assembly, print `Not you, Assembly`.

```
languages = ['C/C++', 'Python', 'R', 'Java', 'SQLs', 'Assembly', 'Go', 'Rust', 'Kotlin']
```

```
# Write your code here
for lang in languages:
    if(lang == "Assembly"):
        print("Not you, Assembly")
    else:
        print(f"I love {lang}!")
```

```
I love C/C++!
I love Python!
I love R!
I love Java!
I love SQLs!
Not you, Assembly
I love Go!
I love Rust!
I love Kotlin!
```

4. Write a code to print every number from 1 to 25 except the one that is divisible by 3.

```
# Write your code here
for i in range(1,26):
    if(i % 3 == 0):
        continue
    print(i)
```

```
1
2
4
5
7
8
10
11
13
14
16
17
19
20
22
23
25
```

5. Write a code that finds the number that is divisible by 7 in a given range.

```python
lower_bound = 1
upper_bound = 100
divisor = 7

result = []
```

```python
# Write your code here
for i in range(lower_bound,upper_bound+1):
    if(i % divisor == 0):
        result.append(i)
print(result)
```

```
[7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98]
```

6. Write a code that construct the following pattern.

input: 5 output: *##### *#### ### *## *****#*

input: 10 output: *########## *######### ######## *####### ###### *##### ***#### ****### *****## **#*

```python
# input: 5
# output:
# *#####
# **####
# ***###
# ****##
# *****#
# Write your code here
n = int(input("Enter a number: "))

for i in range(1, n + 1):

    for j in range(i):
        print("*", end="")

    for k in range(n - i + 1):
        print("#", end="")

    print()
```

```
Enter a number: 10
*##########
**#########
***########
****#######
*****######
******#####
*******####
********###
*********##
**********#
```

5. Functions

1. Define a function `average` that takes arbitrary number of arguments and calculate the mean of input.

```python
# Write your code here
def average(args):
    if len(args) == 0:
        return 0

    mean = sum(args) / len(args)
    return mean

print(average([1, 2, 3, 4, 5]))
```

```
3.0
```

2. Define a function `sumproduct` that takes 2 equal-sized lists and calculate sum of the products of two lists.
It should look like this:

> sumproduct([1,2,3],[4,5,6])
> output: 32

(1 * 4) + (2 * 5) + (3 * 6) = 32

```python
# Write your code here
def sumproduct(list1, list2):
    if len(list1) != len(list2):
        return None

    sum = 0
    for i in range(len(list1)):
        sum += list1[i] * list2[i]
    return sum

print(sumproduct([1, 2, 3], [4, 5, 6]))
```

```
32
```

3. Define a function `fibonacci` that returns Fibonacci number at `n` position.
A Fibonacci number at position `n` is defined by `F(n) = F(n-1) + F(n-2)`. Where `F(0) = 0` and `F(1) = 1`
Example: `0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...`

```python
def fibonacci(n):
    if n == 0:
        print("0")
        return
    elif n == 1:
        print("0, 1")
        return

    dp = [0] * (n + 1)
    dp[0] = 0
    dp[1] = 1

    for i in range(2, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2]

    for i in range(n + 1):
        print(dp[i], end=", " if i < n else "\n")

fibonacci(10)
```

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55
```

4. Define a function `is_palindrome` that takes input string and check whether it is a palindrome or not.
A string is a palindrome if it reads the same forward and backwards.

Example: `madam`, `race car`, `borrow or rob`, `amore roma`, `never odd or even`
Do not consider whitespace. Use `str.replace(' ', '')` to remove whitespace from your string.
Case-insensitive. You can turn everything into lower or uppercase using `str.lower()` or `str.upper()`

Hint: you can reverse the string using `[::-1]` slice.

```python
str1 = "radar" # palindrome
str2 = "rotator" # palindrome
str3 = "lemon" # not palindrome
```

```python
# Write your code here
def is_palindrome(str):
    str = str.replace(" ", "").lower()
    if str == str[::-1]:
        return True
    else:
        return False

print(is_palindrome(str1))
print(is_palindrome(str2))
print(is_palindrome(str3))
```

```
True
True
False
```

5. An `anagram` is a word or phrase formed by rearranging the letters of a different word or phrase.
Define a function `is_anagram` that takes in 2 strings and check whether it is possible to compose a second string using letters in the first string or not.

Example: `Tom Marrvolo Riddle` can be rearraged into `I am Lord Voldermort`
`Meaning of Life` can be rearranged into `Engine of a Film`
Do not consider whitespace. Use `str.replace(' ', '')` to remove whitespace from your string.
Case-insensitive. You can turn everything into lower or uppercase using `str.lower()` or `str.upper()`
Returns only `True` of `False`

```python
# Write your code here
str1 = "Meaning of Life"
str2 = "Engine of a Film"

def is_anagram(str1, str2):
    str1 = str1.replace(" ", "").lower()
    str2 = str2.replace(" ", "").lower()
    if sorted(str1) == sorted(str2):
        return True
    else:
        return False

print(is_anagram(str1, str2))
```

```
True
```
