CSX3001/ITX3001
Fundamentals of Computer Programming

CLASS 10 TO FUNCTION IN PYTHON

PYTHON

USER DEFINED FUNCTIONS

Function is a group of related statements or a code block that perform a specific task. It can be considered as a sub-program.

You have used many pre-defined functions so far. How to create your own function? This part will be the introduction of a user defined function. For more detail, it will be covered in the next session.
The syntax of a function is

```
def functionName(parameters):
    statement #1
    statement #2
    :
    statement #n
```

To create a function, the keyword `def` is specified at the header of function and followed with `functionName` which is the valid identifier. After that it will be followed with parameter(s) in bracket() and end with colon :. The body of function is a code block. It possible to be a statement or many statements.

Example

```
def PrintMyName():
    print("My name is Harry Potter.")
```

Calling a Function

```
def PrintMyName():
    print("My name is Harry Potter.")

PrintMyname()
```

Arguments

Information can be passed into functions as arguments.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma. The following example has a function with one argument (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

Example:

```
def PrintMyName(myname):
   print("My name is" + myname)

PrintMyname("Harry Potter")
PrintMyname("Iron Man")
```

Number of Arguments

By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

Example: This function expects 2 arguments.

```
def PrintNameSurname(firstname, lastname):
   print("My name is" + firstname + " " + lastname)

PrintNameSurname("Harry", "Potter")
PrintNameSurname("Iron", "Man")
```

If you try to call the function with 1 or 3 arguments, you will get an error:

Example:

```
def PrintNameSurname(firstname, lastname):
   print("My name is" + firstname + " " + lastname)

PrintNameSurname("Harry")
```

Arbitrary Arguments, *args

If you do not know how many arguments that will be passed into your function, add a * before the parameter name in the function definition.

This way the function will receive a tuple of arguments, and can access the items accordingly:

```
def PrintCar(*cars):
   print("Cars in stock incldues" + cars[2])

PrintCar("Toyota", "BMW", "Honda", "Volvo")
```

Keyword Arguments

You can also send arguments with the *key = value* syntax.

This way the order of the arguments does not matter. For example,

```
def PrintYoungestKid(kid3, kid2, kid1):
   print("The youngest kid is" + kid3)

PrintYoungestKid(kid1 = "Tom", kid2 = "John", kid3 = "Ann")
```

Return Value(s)

A function can return data as a result. To let a function return a value, use the return statement.

Example:

```
def DoubleValue(x):
   return 2 * x

output = DoubleValue(5)
print(output)               # First output

print(DoubleValue(8))       # Second output
```

Let's take a look at the following Python code that find the sum of even number that a user has entered.

```
intList = [int(intList) for intList in input("Enter multiple
values: ").split()]
total = 0
for i in intList:
    if i % 2 == 0:
        total = total + i # total += i
print(f'The sum of even integer values is {total}')
```

Anytime that you want to find the sum of even numbers (either from a list or user inputs), you have to write that for-loop again. What if we create a function and call it when we want to calculate the sum of even numbers.

```
def FindSumEven(inputList):
    total = 0
    for i in inputList:
        if i % 2 == 0:
            total = total + i # total += i
    return total

intList = [int(intList) for intList in input("Enter multiple
value: ").split()]

output = FindSumEven(intList)

print(f'The sum of all even numbers is {output}')

print(f'The sum of all even numbers is {FindSumEven(intList)}')
```

Based on the above example, write a Python function that find a total sum of all odd numbers that are greater than 5 and less than 20. The function name is FindSumOddwithCondition(inputList)

The following Python function returns multiple values.

```
def FindCountandSumOdd(intList):
    count = 0
    total = 0
    for i in intList:
        if i % 2 == 1:
            total += i
            count += 1
    return total, count

intList = [1,2,3,5,6,7,9,10]

total_output, count_output = FindCountandSumOdd(intList)
print(f'The number of odd integer is {count_output} and the total
summation is {total_output}')
```

◆ FUNCTION EXERCISES

Complete the following exercises in Python IDLE or Jupyter notebook.

1) Write a function, namely MyBio(name, surname, age), that takes three input arguments (name, surname and age). If an age is below 5 or is greater than 70, it prints "Dear {name} {surname}, you should stay home to avoid Covid-19." Otherwise, it prints "Dear {name} {surname}, please stay home as much as you can." Note that this function does not return any values.

   Write a Python code asking for name, surname and age of a user, then call a function and print the output.

2) Write a function, namely DoubleTriple(x), that takes an integer and returns 3 values; a value *x*, double of value *x* and triple of value *x*. Write a Python code to get a value and store in a variable x, then call a function and print the three values on the screen.

3) Write a function, namely FindSumAvg(list_x), that takes a list of values (either integer or floating point values), calculates and returns the summation and average values of all numbers in the list. Then write a Python program to take multiple integer or floating point values and stores in a variable, *list_x* and call FindSumAvg(list_x), and print the summation and average values on the screen.

4) Write a function, namely MergeAndSort(list1, list2), that take two lists as input arguments. This function merges these two lists and returns a list with sorted values in descending order. For example,

   list1 = [1,4,6]
   list2 = [3,7,9]

   outlist = [9,7,6,4,3,1]

5) Write two functions, namely OnlyOdd(intList) and OnlyEven(intList), that takes a list of integers as an input argument. OnlyOdd() function removes all even numbers and returns a list with only odd numbers. OnlyEven() function does the opposite by removing all odd numbers and returning a list of only even numbers.

   Write a Python code to take multiple integer inputs and check if a total number of odd integers is greater than even integers, call OnlyOdd() function. Otherwise, call OnlyEven() function. Then the code prints an output list on a screen. For example,

   Input - [1,2,3,5,7,8,9] ← the number of odd integer is higher
   Output – [1,3,5,7,9]

   Input – [1,2,4,6,8]
   Output – [2,4,6,8]

ASSIGNMENTS

Complete the following exercises in Python IDLE.

1) Given a header of a function,

```
def matrixMultiplication(mat1, mat2):
```

write a complete matrix multiplication function and also test your function by writing a complete code starting from asking a user to specify the size of two matrix and get values of each element via user input.

2) Write a function, namely DistanceConversion(input) that converts a distance from kilometers to miles or vice versa depending on the input. The input must strictly follow the specific format (digitunit, for example 140km or 10mi). If the input is “km”, the output will be “mi”, and vice versa. The function returns both the original and the calculated distance with the corrected unit. The results will be printed outside the function. For example,

```
Enter a distance (in km or mi): 10mi
10mi is equal to 16km.

Enter a distance (in km or mi): 32km
32km is equal to 20mi.
```