

Submitted By	Hooria Khan Orakzai
ID	14263
Department	BS(Software-Engineering)
Semester	5th
Submitted To	Sir Ayub

Department of Computer Science
Final Exam Spring 2020

Subject: Data Sciences

BS (CS,SE)

Instructor: M.Ayub Khan

Note:

At the top of the answer sheet there must be the ID, Name and semester of the concerned Student.

Students must have to provide the output of their respective programs. Students have same answers or programs will be considered fail. Programs in Python and codes should be explained clearly.

As this assignment is online so incase of any ambiguity my Whatsapp no. is 034499121116.

Q1. a. Why Functions are used discuss in detail?

Ques 1 Part(A)

Functions are use in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or cannot return one or more values.

Types of Function:

There are three types of functions in Python:

- 1) Built-in functions, such as `help()` to ask for help, `min()` to get the minimum value, `print()` to print an object to the terminal.
- 2) User-Defined Functions (UDFs), which are functions that users create to help them out
- 3) Anonymous functions, which are also called lambda functions because they are not declared with the standard `def` keyword

b. How arguments are used in function, write a simple program in Python?

Ques 1 Part(B)

- We can write information in a function as an argument. Arguments are used after function we can use 0 to infinity arguments in print function and just separate them with comma (,)
- Just one argument is used in type function. Type function only display the datatype of argument.



- Some people think arguments and parameters are same but they are different.
- Parameters and Arguments are used for same thing that the information are passed in to a function.

Difference is

- ❖ A parameter is a variable in a method definition. When a method is called, the arguments are the data you pass into the method's parameters. Parameter is variable in the declaration of function.
- ❖ Argument is the actual value of this variable that gets passed to function.

```
In [3]: # Defining Function
def printinfo( name, age ): #Here we are using two arguments which are (name) and (Age)

    print ("Name: ", name); #print is function
    print ("Age ", age);
    return;

# Now you can call printinfo function
printinfo( age=23, name="hooria" ); #if we don't use two arguments then the programme gave error

Name: hooria
Age 23

In [ ]:
```

Explanation Of Code:

In this program I have use print function because it can accept 0 to infinity arguments and here am using two arguments age and name and and I have put values in function when I recall printinfo function it shows Name : hooria and age 23.

Q2. a. Why .upper(),.lower(),capitalize() and .swapcase() function are used ?

Ques 2 Part(A)

1) Upper Function:

Upper function are used to convert the all the lowercase letters in to upper case letter.

Syntax for Upper Case is my_string.isupper()

2) Lower Function:

Lower Function are used to convert all the upper-case letters in to lowercase

Syntax for Lower Case is my_string.islower()

3) Capitalize Function:

Capitalize Function it converts the first letter of word in to capital letter and remaining letters of word will be converted in to small letter

Syntax for capitalize function is my_string.capitalize()

4) Swapcase Function:

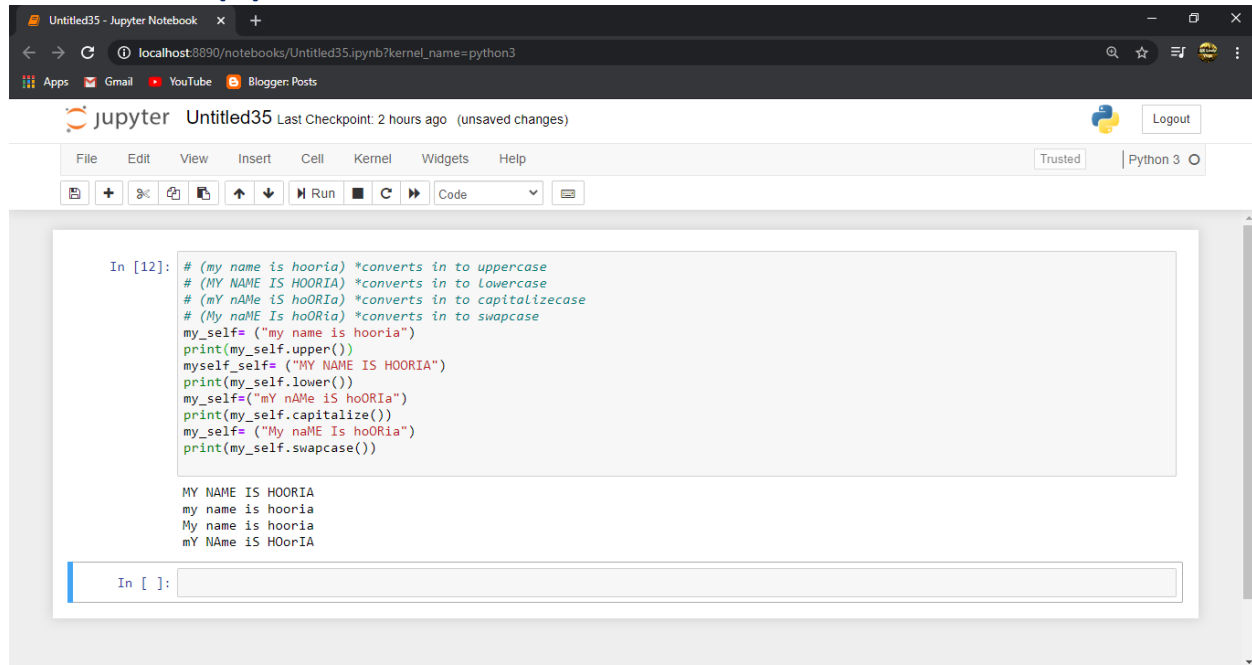
Swapcase function it convert all uppercase letters in to lowercase and lowercase letter in to uppercase letter.

Syntax for swapcase function is `my_string.swapcase()`

b. Write a program in which the discussed functions are used.

Note : Q2 part a functions.

Ques 2 Part(B)



```
In [12]: # (my name is hooria) *converts in to uppercase
# (MY NAME IS HOORIA) *converts in to lowercase
# (mY nAMe iS hoORiA) *converts in to capitalizcase
# (My naME Is hoORia) *converts in to swapcase
my_self= ("my name is hooria")
print(my_self.upper())
myself_self= ("MY NAME IS HOORIA")
print(my_self.lower())
my_self=("mY nAMe iS hoORiA")
print(my_self.capitalize())
my_self= ("My naME Is hoORia")
print(my_self.swapcase())

MY NAME IS HOORIA
my name is hooria
My name is hooria
mY NAME iS HOoRiA
```

Explanation of Code:

In this programme we convert lowercase in to uppercase, uppercase in to lowercase, capitalize function in which the first letter of word convert in to uppercase and rest of the letters of a word in to lowercase, swapcase upper case letters convert in to lowercase and lowercase convert in to uppercase. The programme shows all the execution of the given function and how these functions work

Q3. a. What are the rules for defining the function?

Ques 3 Part(A)

The following are some rules for defining function.

- ❖ the function must be defined prior to the function call or a stack trace will occur.
- ❖ Every function has a signature. This signature is the name, parameters and the data types being passed in and out.
- ❖ Function blocks begin with the keyword `def` followed by the function name and parentheses (()).

- ❖ Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- ❖ The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.
- ❖ The code block within every function starts with a colon (`:`) and is indented
- ❖ The first statement of a function can be an optional statement - the documentation string of the function

Use the `def` statement when creating a function

Syntax:

```
def functionname( parameters ):
```

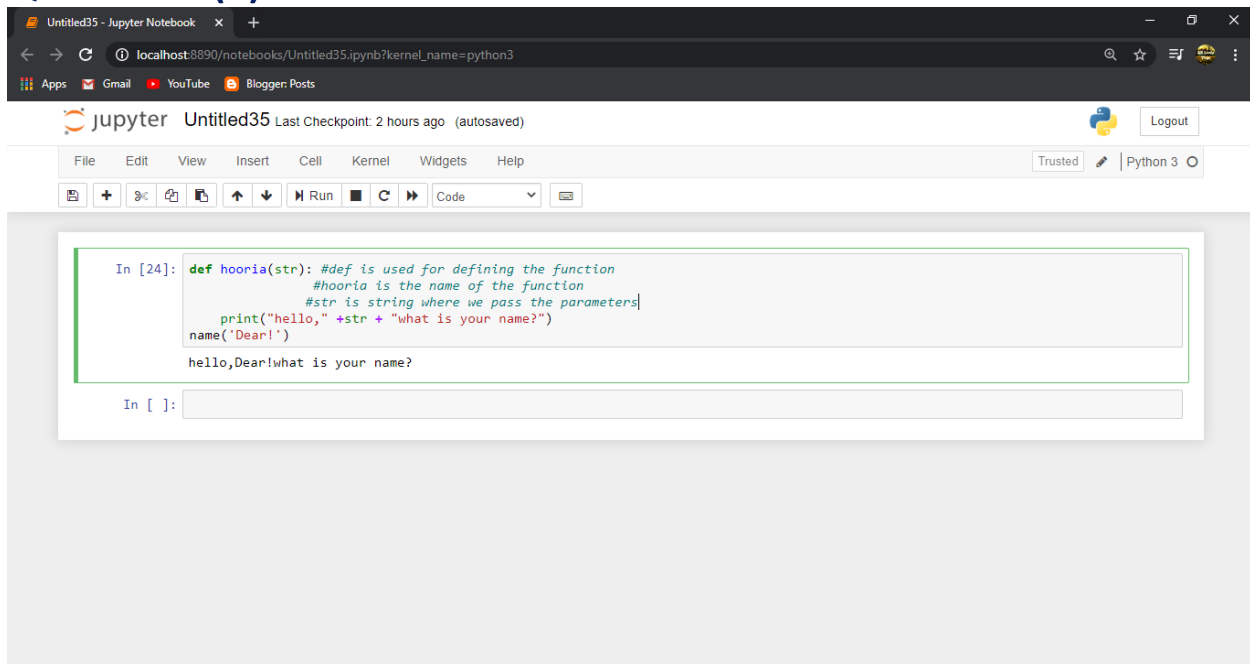
```
    "function_docstring"
```

```
    function_suite
```

```
    return [expression]
```

b. Write a suitable program of our defined function in Python?

Ques 3 Part(B)



```

Untitled35 - Jupyter Notebook
localhost:8890/notebooks/Untitled35.ipynb?kernel_name=python3
jupyter Untitled35 Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [24]: def hooria(str): #def is used for defining the function
          #hooria is the name of the function
          #str is string where we pass the parameters
          print("hello," +str + "what is your name?")
          name("Dear!")
          hello, Dear! what is your name?
In [ ]:

```

EXPLANATION of Code:

In this function, the parameter (`str`) does not have a default value and is required (mandatory) during a call. On the other hand, the

parameter "hello" has a default value of "WHAT IS YOUR NAME!". So, it is optional during a call. If a value is provided, it will overwrite the default value. Any number of arguments in a function can have a default value. But once we have a default argument, all the arguments to its right must also have default values.

Q4. a. What are the rules for defining the function and Parameter passing to the function?

Ques 4 Part(A)

- ❖ If you **pass** immutable **arguments** like integers, strings or tuples to a **function**, the **passing** acts like call-by-value. The object reference is **passed** to the **function parameters**
- ❖ Def keyword is used to define custom function or method the arguments of the functions are given between parentheses followed by a colon then the function body and finally the return objects for optionally returning values.

Parameter passing to the function:

Call by Value

The most common strategy is the call-by-value evaluation, sometimes also called pass-by-value. Pass by Value. Pass by Value, means that a copy of the data is made and stored by way of the name of the parameter So, if the expression is a variable, a local copy of its value will be used, i.e. the variable in the caller's scope will be unchanged when the function returns.

Call by Reference:

- In call-by-reference evaluation, which is also known as pass-by-reference, a function gets an implicit reference to the argument, rather than a copy of its value. As a consequence, the function can modify the argument, i.e. the value of the variable in the caller's scope can be changed. The advantage of call-by-reference consists in the advantage of greater time- and space-efficiency, because arguments do not need to be copied. A reference parameter "refers" to the original data in the calling function.

Syntax:

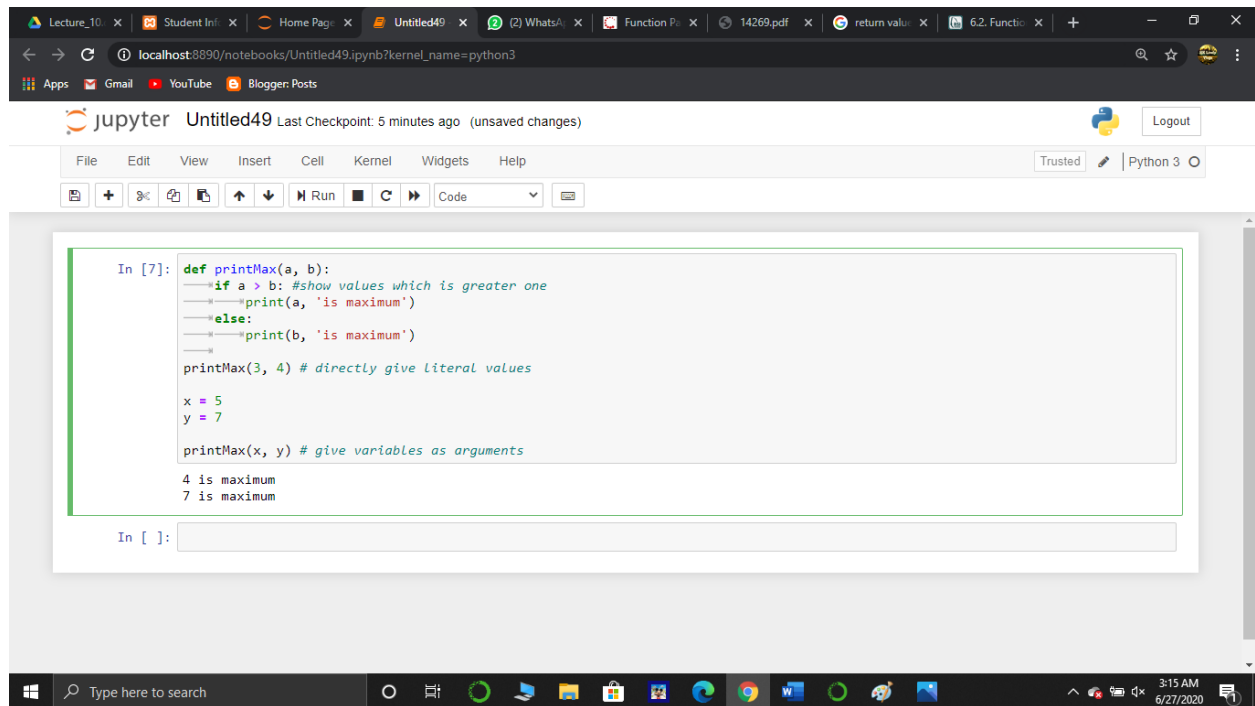
```
def ref_demo(x):  
    print "x=",x," id=",id(x)
```

```
x=42
```

```
print "x=",x," id=",id(x)
```

b. Write a suitable program of our defined function by parameter passing in Python?

Ques 4 Part(B)



```
In [7]: def printMax(a, b):
         #show values which is greater one
         print(a, 'is maximum')
         else:
         print(b, 'is maximum')
         printMax(3, 4) # directly give literal values
         x = 5
         y = 7
         printMax(x, y) # give variables as arguments
         4 is maximum
         7 is maximum
```

Explanation of Code:

we define a function called printMax where we take two parameters called a and b. We find out the greater number using a simple if..else statement and then print the bigger number. In the first usage of printMax, we directly supply the numbers i.e. arguments. In the second usage, we call the function using variables. printMax(x, y) causes value of argument x to be assigned to parameter a and the value of argument y assigned to parameter b. The printMax function works the same in both the cases.

Q5. a. What are return values to a Function discuss in detail?

Ques 5 Part(A)

- ❖ A return is a value that a function returns to the calling script or function when it completes its task. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs.
- ❖ You use the Function Returns edit combo box in the General page of the New Script dialog to tell HOORIA the type of value the function returns. You also type the description of the return in the Return description edit box. Adding a description for the return, helps you and anyone else using your function to determine exactly what the value should be used for within the calling script or function.
- ❖ When you create a new function that returns a string value, the Script Manager places the following function beginning line into your script file:
String Function MyFunction ()
- ❖ The "string" key word that precedes the "function" key word tells you that the MyFunction function returns a string value to the calling script or user-defined function.

b. Write a suitable program of a Function with returning value?

Ques 5 Part(B)

Untitled48 - Jupyter Notebook x +

localhost:8890/notebooks/Untitled48.ipynb?kernel_name=python3

Apps Gmail YouTube Blogger: Posts

jupyter Untitled48 Last Checkpoint: 9 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [3]: def fun(): #fun is function
        str = "hooria" #str is tstring
        x = 23 #give value to x
        return str, x; # Return string, and x we could also
           # write (str, x)

#code to test above method
str, x = fun() # here we are calling return value outside function
print(str) #print str and x seprately
print(x)

hooria
23
```

In []: