J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

7. PYTHON FUNCTIONS

Section – A

Choose the best answer **(1 Mark)**

1. A named blocks of code that are designed to do one specific job is called as
   (a) Loop (b) Branching **(c) Function** (d) Block
2. A Function which calls itself is called as
   (a) Built-in **(b) Recursion** (c) Lambda (d) return
3. Which function is called anonymous un-named function
   (a) Lambda (b) Recursion (c) Function (d) define
4. Which of the following keyword is used to begin the function block?
   (a) define (b) for (c) finally **(d) def**
5. Which of the following keyword is used to exit a function block?
   (a) define **(b) return** (c) finally (d) def
6. While defining a function which of the following symbol is used.
   (a) ; (semicolon) (b) . (dot) **(c) : (colon)** (d) $ (dollar)
7. In which arguments the correct positional order is passed to a function?
   (a) Required (b) Keyword (c) Default (d) Variable-length
8. Read the following statement and choose the correct statement(s).

(I) In Python, you don't have to mention the specific data types while defining function.

(II) Python keywords can be used as function name.

(a) I is correct and II is wrong

(b) Both are correct

(c) I is wrong and II is correct

(d) Both are wrong

9. Pick the correct one to execute the given statement successfully.

if ____ : print(x, " is a leap year")

(a) x%2=0 **(b) x%4==0** (c) x/4=0 (d) x%4=0

10. Which of the following keyword is used to define the function testpython(): ?
    (a) define (b) pass **(c) def** (d) while

Section-B

Answer the following questions **(2 Marks)**

1. What is function?

- Functions are named blocks of code that are designed to do one specific job.
- Types of Functions are User defined, Built-in, lambda and recursion.
- Function blocks begin with the keyword "def " followed by function name and parenthesis ().

2. Write the different types of function.

TYPES OF FUNCTION:

3. What are the main advantages of function?

- **Main advantages of functions are ,**
  - It avoids repetition and makes high degree of code reusing.
  - It provides better modularity for your application.

4. What is meant by scope of variable? Mention its types.

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
- Scope holds the current set of variables and their values.
- The two types of scopes are- **local scope** and **global scope**.

5. Define global scope.

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

6. What is base condition in recursive function

- A recursive function calls itself.
- The condition that is applied in any recursive function is known as base condition.
- A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

7. How to set the limit for recursive function? Give an example.

- Python stops calling recursive function after 1000 calls by default.
- So, It also allows you to change the limit using sys.setrecursionlimit (limit_value).
- **Example:**

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
if n == 0:
return 1
else:
return n * fact(n-1)
print(fact (2000))
```

Section-C

Answer the following questions **(3 Marks)**

1. Write the rules of local variable.

- A variable with local scope can be accessed only within the function/block that it is created in.
- When a variable is created inside the function/block, the variable becomes local to it.
- A local variable only exists while the function is executing.
- The formal arguments are also local to function.

2. Write the basic rules for global keyword in python.

The basic rules for ***global*** keyword in Python are:

- When we define a variable outside a function, it's global by default. You don't have to use global keyword.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect.

3. What happens when we modify global variable inside the function?

- If we modify the global variable , We can see the change on the **global** variable outside the function also.

Example:

```
x = 0                                   # global variable
def add():
    global x
    x = x + 5                           # increment by 2
  print ("Inside add() function x value is :", x)
add()
print ("In main x value is :", x)
```

Output:

```
Inside add() function x value is : 5
In main x value is : 5                  #value of x changed outside the
    function
```

4. Differentiate ceil() and floor() function?

ceil()	floor()
Returns the smallest integer greater than or equal to x	Returns the largest integer less than or equal to x
math.ceil (x)	math.floor (x)

5. Write a Python code to check whether a given year is leap year or not.

CODE:

```
n=int(input("Enter the year"))
if(n%4==0):
   print ("Leap Year")
else:
   print ("Not a Leap Year")
```

Output:

```
Enter the year      2012
Leap Year
```

6. What is composition in functions?

- The value returned by a function may be used as an argument for another function in a nested manner.
- This is called **composition**.
- **For example,** if we wish to take a numeric value as a input from the user, we take the input string from the user using the function **input()** and apply **eval()** function to evaluate its value.

7. How recursive function works?

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

8. What are the points to be noted while defining a function?

When defining functions there are multiple things that need to be noted;

- Function blocks begin with the keyword **"def"** followed by function name and parenthesis ().
- Any input parameters should be placed within these parentheses.
- The code block always comes after a colon (:) and is indented.
- The statement **"return [expression]"** exits a function, and it is optional.
- A **"return"** with no arguments is the same as return None.

Section - D

Answer the following questions: **(5 Marks)**

1. Explain the different types of function with an example.

- Functions are named blocks of code that are designed to do one specific job.
- **Types of Functions**
  - User defined Function
  - Built-in Function
  - Lambda Function
  - Recursion Function

i) BUILT-IN FUNCTION:

- Built-in functions are Functions that are inbuilt with in Python.
- print(), echo() are some built-in function.

ii) USER DEFINED FUNCTION:

- Functions defined by the users themselves are called user defined function.
- Functions must be defined, to create and use certain functionality.
- Function blocks begin with the keyword "def " followed by function name and parenthesis ().
- When defining functions there are multiple things that need to be noted;
  - Function blocks begin with the keyword **"def"** followed by function name and parenthesis ().
  - Any input parameters should be placed within these parentheses.
  - The code block always comes after a colon (:) and is indented.
  - The statement **"return [expression]"** exits a function, and it is optional.
  - A **"return"** with no arguments is the same as return None.
- **EXAMPLE:**

```
def area(w,h):
    return w * h
print (area (3,5))
```

iii) LAMBDA FUNCTION:

- In Python, anonymous function is a function that is defined without a name.
- While normal functions are defined using the **def** keyword, in Python anonymous functions are defined using the **lambda** keyword.
- Hence, anonymous functions are also called as **lambda** functions.

USE OF LAMBDA OR ANONYMOUS FUNCTION:

- Lambda function is mostly used for creating small and one-time anonymous function.
- Lambda functions are mainly used in combination with the functions like filter(), map() and reduce().

EXAMPLE:

```
sum = lambda arg1, arg2: arg1 + arg2
print ('The Sum is :', sum(30,40))
```

```
print ('The Sum is :', sum(-30,40))
```

Output:

The Sum is : 70

The Sum is : 10

iv) RECURSIVE FUNCTION:

Functions that calls itself is known as recursive.

Overview of how recursive function works

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

2. Explain the scope of variables with an example.

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
- We can say that scope holds the current set of variables and their values.
- There are two types of scopes - **local scope** and **global scope**.

➢ **Local Scope:**

- A variable declared inside the function's body or in the local scope is known as local variable.

Rules of local variable:

- A variable with local scope can be accessed only within the function/block that it is created in.
- When a variable is created inside the function/block, the variable becomes local to it.
- A local variable only exists while the function is executing.
- The formal arguments are also local to function.

Example:

```
def loc():
y=0 # local scope
print(y)
loc()
```

Output:

0

- **Global Scope**

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

- **Rules of global Keyword**

The basic rules for ***global*** keyword in Python are:

- When we define a variable outside a function, it's global by default. You don't have to use global keyword.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect

Use of global Keyword

- Without using the global keyword we cannot modify the global variable inside the function but we can only access the global variable.

Example:

```
x = 0                                          # global variable
def add():
global x
x = x + 5                                      # increment by 2
print ("Inside add() function x value is :", x)
add()
print ("In main x value is :", x)
```

Output:

```
Inside add() function x value is : 5
In main x value is : 5                         #value of x changed outside the function
```

3. Explain the following built-in functions.

(a) id() (b) chr() (c) round() (d) type() (e) pow()

Function	Description	Syntax	Example
id ( )	Return the "identity" of an object. i.e. the address of the object in memory.	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) **Output:** address of x is : 1357486752 address of y is : 13480736
chr ( )	Returns the Unicode character for the given ASCII value.	chr (i)	c=65 print (chr (c)) **Output:** A
round ( )	Returns the nearest integer to its input. 1. First argument (number) is used to specify the value to be rounded.	round (number [,ndigits])	x= 17.9 print ('x value is rounded to', round (x)) **Output:** X value is rounded to 18
type ( )	Returns the type of object for the given single object.	type (object)	x= 15.2 print (type (x)) **Output:** <class 'float'>
pow ( )	Returns the computation of a,b i.e. (a**b) a raised to the power of b.	pow (a,b)	a= 5 b= 2 print (pow (a,b)) **Output:** 25

4. Write a Python code to find the L.C.M. of two numbers.

CODE:

```
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
if x>y:
    min=x
else:
    min=y
while(1):
  if((min%x == 0) and (min % y == 0)):
      print("LCM is:",min)
      break
  min=min+1
```

OUTPUT:

```
Enter first number:2
Enter second number:3
LCM is: 6
```

5. Explain recursive function with an example.

- Functions that calls itself is known as recursive.
- When a function calls itself is known as recursion.
- Recursion works like loop but sometimes it makes more sense to use recursion than loop.
- Imagine a process would iterate indefinitely if not stopped by some condition is known as infinite iteration.
- The condition that is applied in any recursive function is known as base condition.
- A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.
- Python stops calling recursive function after 1000 calls by default.
- So, It also allows you to change the limit using sys.setrecursionlimit (limit_value).

Overview of how recursive function works:

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

EXAMPLE:

```
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact (n-1)
    print (fact (0))
    print (fact (5))
```

Output:

1

120

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com