1. In Python, what is the difference between a built-in function and a user-defined function? Provide an example of each.

Ans:-The functions which come along with python by default are called as built in function.The functions which a user creates according to their needs are called user defined functions.

Eg. i) print()   built in function
     ii) def greetings():   user define function
            print("hello")
         greetings()

2. How can you pass arguments to a function in Python? Explain the difference between positional arguments and keyword arguments.

ANS:-Arguments are specified after the function name, inside the parentheses.
 keyword arguments you will get the correct output because the order of argument doesn't matter provided the logic of your code is correct. But in the case of positional arguments, you will get more than one output on changing the order of the arguments.

3. What is the purpose of the return statement in a function? Can a function have multiple return statements? Explain with an example.

ANS:-A return statement ends the processing of the current function and returns control to the caller of the function.yes a function can have multiple return statement but the first one to satisfy the condition will get executed.

```python
def type_of_int(number):
    if number % 2 == 0:
        return 'even'
    else:
        return 'odd'


result = type_of_int(7)
print(result)
```

4. What are lambda functions in Python? How are they different from regular functions? Provide an example where a lambda function can be useful.

ANS:- Lambda functions are single line functions,and it is not necessary to start a lambda function with the " def " keyword ,but the regular functions have to start with def keyword.

```python
add =lambda x,y:x+y
print(add(4,6))
```

5. How does the concept of "scope" apply to functions in Python? Explain the difference between local scope and global scope.

Ans:-A variable is only available from inside the region it is created. This is called scope.A variable created inside a function belongs to the *local scope* of that function, and can only be used inside that function.A variable created in the main body of the Python code is a global variable and belongs to the global scope.Global variables are available from within any scope, global and local.

6. How can you use the "return" statement in a Python function to return multiple values?
ANS:-You can use a return statement to return multiple values from a function. To do that, you just need to supply several return values separated by commas.

7. What is the difference between the "pass by value" and "pass by reference" concepts when it comes to function arguments in Python?
ANS:-The difference between pass-by-reference and pass-by-value is that modifications made to arguments passed in by reference in the called function have effect in the calling function, whereas modifications made to arguments passed in by value in the called function can not affect the calling function.

8. Create a function that can intake integer or decimal value and do following operations:
a. Logarithmic function (log x)
b. Exponential function (exp(x))
c. Power function with base 2 (2x)
d. Square root
Ans:-

```python
import math
print("ENTER THE CHOICE IN LOWERCASE ")
choice = str(input("select your choice:log/expo/power/sqrt ::"))
x = float(input("Enter the number:"))
def calculation():
    if choice == "log":
        print("The logarithmic function of log",x,"is:",math.log(x,10))

    elif choice == "expo":
        print("The exponential function of ",x,"is:",math.exp(x))

    elif choice == "power":
        print("The power function with base2 of",x,"is:",math.pow(2,x))

    elif choice == "sqrt":
        print("The Square root function of ",x,"is:",math.sqrt(x))


calculation()
```

9. Create a function that takes a full name as an argument and returns first name and last name.
ANS:-

```python
def split():
    full_name = input("enter your name :")
    a= full_name.split()
    print(f"first name is{a[0]} and last name is {a[1]}")
split()
```