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

Sol- built in function- standard library

Eg- print() and

user- defined - created by the user

Eg- def shape()

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

Sol- Positional arguments- It is passed based on their position or order**.**

Keyword arguments- means arguments by names

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

Sol- To specify values that function should produce as its result.

Yes, function can have multiple returns

```python
def is_even_or_odd(number):
    if number % 2 == 0:
        return "Even"
    else:
        return "Odd"

num1 = 10
num2 = 7

result1 = is_even_or_odd(num1)
result2 = is_even_or_odd(num2)

print(num1, "is", result1)
print(num2, "is", result2)
```

4. What **are** lambda **functions in** Python? **How are** they **different from** regular **functions? Provide** an

example where **a** lambda **function can be useful**.

Sol- which does not have a function name and is defined by using keyword lambda.

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

Sol- Local scope- variables which are accessible within that block or function.

Global scope- that are defined outside of any block or function and are accessible throughout the program.

6. How can **you use the** "return**" statement in** a Python function to **return** multiple **values?**

Sol- Using comma separated values.

7**. What is the** difference between the **"pass by value" and** "**pass by** reference**" concepts when** it

comes to function arguments in Python?

Sol- In pass by value a copy of the value of a variable is passed to the 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 (2×)**

d. Square **root**

Sol-

```
import math

def math_operations(x):
    logarithmic = math.log(x)
    exponential = math.exp(x)
    power_of_two = math.pow(2, x)
    square_root = math.sqrt(x)

    return logarithmic, exponential, power_of_two, square_root

# Test the function
value = 5.0
log_val, exp_val, pow_val, sqrt_val = math_operations(value)

print("Logarithmic function (log x):", log_val)
print("Exponential function (exp(x)):", exp_val)
print("Power function with base 2 (2^x):", pow_val)
print("Square root:", sqrt_val)
```

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