Chapter 8 – Functions & Recursions.

Introduction to Functions

A function is a group of statements that perform a specific task. As programs grow in size and complexity, functions help keep the code organized and manageable. Functions can be reused any number of times within a program, making code more modular and efficient.

Function Syntax and Example

The syntax of a function looks as follows:

```
def func1():
    print('hello')
```

This function can be called any number of times, anywhere in the program.

Function Call

Whenever we want to call a function, we use the function name followed by parentheses:

```
func1()  # This is a function call.
```

Function Definition

The part containing the exact set of instructions executed during the function call is known as the function definition.

Example: Greet a User

```
def greet():
    print("Good day")
greet()
```

Types of Functions in Python

1. **Built-in Functions**: Functions that are already present in Python, such as `len()`, `print()`, `range()`, etc.
2. **User-defined Functions**: Functions defined by the user, such as `func1()`.

Functions with Arguments

A function can accept values (arguments) that it can work with. These values are placed in the parentheses.

```
def greet(name):
    greeting = "Hello " + name
    return greeting

a = greet("Shivam")
print(a)  # Output: Hello Shivam
```

Default Parameter Value

A function can have default parameter values. If no argument is passed, the default value is used.

```
def greet(name="stranger"):
    print("Hello " + name)

greet()        # Output: Hello stranger
greet("Shivam") # Output: Hello Shivam
```

Recursion

Recursion is a technique where a function calls itself. It is useful for directly implementing mathematical formulas as functions.

Example: Factorial Function

```python
def factorial(n):
    if n == 0 or n == 1:  # Base condition to stop recursion
        return 1
    else:
        return n * factorial(n - 1)  # Recursive call
```

Explanation of Recursion

- **Base Condition**: The condition at which the function stops calling itself, preventing infinite recursion.
- **Recursive Call**: The function calls itself with a different argument.

Example of Usage:

```python
result = factorial(5)
print(result)  # Output: 120
```

Important Notes on Recursion

The programmer needs to be extremely careful while working with recursion to ensure that the function does not infinitely call itself. Recursion is sometimes the most direct way to code an algorithm, but it must be used with caution.