Name: Katricia Lopez
Sec:CS3C

Defining a Function

In Python, a function is a block of reusable code that performs a specific task. It's defined using the `def` keyword followed by the function name and parentheses, optionally followed by parameters and a colon.

Reasons for Using Functions

Encapsulation: Functions allow you to encapsulate a piece of code, making it reusable and easier to manage.

Modularity: Breaking down a program into smaller functions makes it easier to understand and maintain.

Abstraction: Functions hide the implementation details, allowing you to focus on the high-level functionality.

Code Reusability: Once defined, functions can be called multiple times from different parts of the program.

Types of Functions in Python

Built-in functions: Functions that are built into Python, such as `print()`, `len()`, etc.

User-defined functions: Functions defined by the user to perform specific tasks.

Advantages of User-Defined Function

Reusability: User-defined functions can be reused in different parts of the program.

Modularity: Breaking down the code into smaller functions makes it easier to understand and maintain.

Abstraction: Functions hide the implementation details, making the code easier to comprehend.

Encapsulation: Functions encapsulate a block of code, reducing code duplication.

Rules for Declaring a Function in Python

Function name should be descriptive and follow Python naming conventions.

Function names cannot start with a number or contain special characters except underscore (_).

Parameters can be optional by providing default values.

The `def` keyword is used to define functions.

Indentation is crucial to define the scope of the function.

Python Function Syntax

```python
def function_name(parameters):
    # Function body
    statements
    return value
```

Function Argument and Parameter

Parameters are variables listed inside the parentheses in the function definition.

Arguments are the values passed into the function when it is called.

The Return Statement

The `return` statement is used to return a value from a function.

If the `return` statement is not present, the function returns `None` by default.

Multiple values can be returned as a tuple.