Panganiban, John Paolo D.
CS3C

## FUNCTIONS

● Defining a Function

- A function in programming is a named block of code that performs a specific task. It typically takes inputs (parameters), processes them, and returns an output.

● Reasons of Using Functions

- Functions in programming offer modularity by breaking down code into manageable parts, promoting reuse of code segments for efficiency, abstracting implementation details to focus on functionality, organizing code into logical units, facilitating debugging by isolating specific tasks, and aiding in testing individual components. Overall, functions enhance code readability, maintainability, and efficiency, making programming tasks more manageable and effective.

● Types of Functions in Python

-Python includes built-in functions, user-defined functions, lambda functions, recursive functions, and anonymous functions.

● Advantages of User – Defined Function

- User-defined functions offer modularity, readability, code reuse, simplification of debugging, and abstraction.

● Rules in Declaring a Function in Python

-In Python, function declaration starts with the 'def' keyword followed by the function name and parameters, if any.

● Python Function Syntax

-This structure defines a function named **function_name** with optional parameters.

- Function Argument and Parameter

-Arguments are the actual values passed to a function during a call, while parameters are the variables listed in the function definition to receive those arguments.

- The Return Statement

- The return statement in a function is used to exit the function and optionally return a value back to the caller.