Ramila Jane B. Abobo                              Finals Assignment #1

BSCS-3C

Functions

1. Defining a Function

- It is a block of statement to accomplish the specific task. The work of function is instead of writing the code repeatedly in a different, we can use the function by calling function name to reuse over and over again.

2. Reasons of Using Functions

- It makes the program understandable and also reduces the redundancy because it can use the function name repeatedly. Also to avoid the confusion in writing the program.

3. Types of Functions in Python

- Built-in Function- It is type of function that built in python itself and available for use. Example is range(), print(), ,min().
- User-defined functions- It is type of function which is the user who will defined the function name to perform a specific task.

4. Advantages of User – Defined Function

- First is Reusability, when you define the function you can repeatedly use the code without using the same logic.
- Second is Readability, your code can be easier to comprehend and more self-explanatory by using well-named functions with descriptive, clear names.
- Functions enable you to divide a difficult issue into smaller, more manageable portions.

5. Rules in Declaring a Function in Python

- In Python, it defined the def keyboard the function, followed by the function name and the parenthesis which hold any parameters the function. Also use a colon (:) to indicate the beginning of the function body. Then the function body and a return statement.

## 6. Python Function Syntax

```python
def function_name(parameters):
    # Statements
    return value   # Optional
```

## 7. Function Argument and Parameter

- Parameter is a variable in a function's declaration

```python
def greet(name):
 print("Hello,", name)
```

- An argument is the actual value that is passed to the function when it is called.

```python
greet("Alice")
```

## 8. The Return Statement

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

```python
def add(x, y):
    return x + y
# Call the function and store the result in a variable
result = add(5, 5)


# Print the result
print(result)  # Output: 10
```