Blank Side

AIM:

To implement Python basics covering variables, print statements, comments, data types (numeric types, string, list, tuple), control structures (if, for, while), and functions (definition, call, return statement, default parameters).

SOFTWARE REQUIRED:

Python 3.11 (or the latest version available for your system).

FLOWCHART:

1. **Start**
2. **Python Basics**
   - Declare Variables
   - Print Statements
   - Comments
3. **Data Types Examples**
   - Integers
   - Floats
   - Strings
   - Lists
   - Tuples
4. **Control Structures**
   - If Statements
   - For Loop
   - While Loop
5. **Functions**
   - Function Definition
   - Function Call
   - Return Statement
   - Default Parameters
6. **End**

Error! Filename not specified.

CODE:

Python Basics

```
python
Copy code
# Python Basics: Variables, Print Statement, and Comments
name = "Alice"
age = 25
height = 5.6

print("Hello, World!")
print("Name:", name)
print("Age:", age)
print("Height:", height)

# This is a single-line comment

'''
This is a
multi-line comment
'''

"""
This is another
multi-line comment
"""
```

Data Types

```
python
Copy code
# Numeric Types
integer_value = 10
float_value = 20.5
print("Integer Value:", integer_value)
print("Float Value:", float_value)

# String
string_value = "Hello, Python!"
print("String Value:", string_value)

# List
list_value = [1, 2, 3, 4, 5]
print("List Value:", list_value)

# Tuple
tuple_value = (1, 2, 3, 4, 5)
print("Tuple Value:", tuple_value)
```

Control Structures

```
python
Copy code
# If Statements
number = 10
if number > 0:
    print("The number is positive.")
elif number == 0:
    print("The number is zero.")
else:
    print("The number is negative.")

# For Loop
for i in range(5):
    print("Iteration:", i)

# While Loop
count = 0
while count < 5:
    print("Count:", count)
    count += 1
```

Functions in Python

```
python
Copy code
# Function Definition
def greet(name):
    print("Hello,", name)

# Function Call
greet("Alice")

# Return Statement
def add(a, b):
    return a + b

result = add(5, 3)
print("Sum:", result)

# Default Parameters
def greet(name="Guest"):
    print("Hello,", name)

greet()  # Uses default parameter
greet("Bob")  # Uses passed argument
```

OUTPUT:

```
yaml
Copy code
Hello, World!
Name: Alice
Age: 25
Height: 5.6
Integer Value: 10
Float Value: 20.5
String Value: Hello, Python!
List Value: [1, 2, 3, 4, 5]
Tuple Value: (1, 2, 3, 4, 5)
The number is positive.
Iteration: 0
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
Hello, Alice
Sum: 8
Hello, Guest
Hello, Bob
```

Ruled Side

Python Basics

Introduction:

Python basics include understanding variables, print statements, and comments. Variables store data, print statements display output, and comments document the code.

AIM:

To implement Python basics covering variables, print statements, and comments.

SOFTWARE REQUIRED:

Python 3.11 (or the latest version available for your system).

ALGORITHM:

1. **Start**
2. Declare and initialize variables
3. Use print statements to display variable values
4. Add comments to explain the code
5. **End**

STEPS:

1. Start the program.
2. Declare variables (`name`, `age`, `height`) and assign values.
3. Use `print` statements to display the values of the variables.
4. Add single-line and multi-line comments to explain the code.
5. End the program.

Data Types

Introduction:

Python supports various data types, including numeric types (integers and floats), strings, lists, and tuples. These data types help in storing and manipulating different kinds of data.

AIM:

To implement and demonstrate different data types in Python.

SOFTWARE REQUIRED:

Python 3.11 (or the latest version available for your system).

ALGORITHM:

1. **Start**
2. Demonstrate numeric data types (integers and floats)
3. Demonstrate string data type
4. Demonstrate list data type
5. Demonstrate tuple data type
6. **End**

STEPS:

1. Start the program.
2. Declare and initialize integer and float variables, then print them.
3. Declare a string variable and print it.
4. Declare a list and print it.
5. Declare a tuple and print it.
6. End the program.

Control Structures

Introduction:

Control structures in Python include if statements and loops (for and while). These structures control the flow of the program based on conditions and repetitive tasks.

AIM:

To implement control structures in Python, including if statements, for loops, and while loops.

SOFTWARE REQUIRED:

Python 3.11 (or the latest version available for your system).

ALGORITHM:

1. **Start**
2. Implement if statements to check conditions
3. Implement for loop for fixed iterations
4. Implement while loop for iterations based on condition
5. **End**

STEPS:

1. Start the program.
2. Use if statements to check if a number is positive, negative, or zero, and print the result.
3. Use a for loop to iterate from 0 to 4 and print each iteration.
4. Use a while loop to count from 0 to 4 and print each count.
5. End the program.

Functions in Python

Introduction:

Functions in Python are reusable blocks of code that perform a specific task. They can take parameters, return values, and have default parameters.

AIM:

To implement functions in Python, including function definition, function call, return statement, and default parameters.

SOFTWARE REQUIRED:

Python 3.11 (or the latest version available for your system).

ALGORITHM:

1. **Start**
2. Define functions with parameters and return values
3. Call functions and display results
4. Use default parameters in functions
5. **End**

STEPS:

1. Start the program.
2. Define a function `greet` that takes a parameter `name` and prints a greeting.
3. Call the `greet` function with the argument `"Alice"`.
4. Define a function `add` that takes two parameters, adds them, and returns the result.
5. Call the `add` function with arguments `5` and `3`, store the result in `result`, and print the sum.
6. Define a function `greet` with a default parameter `name="Guest"`.
7. Call the `greet` function without arguments to use the default parameter, then call it again with the argument `"Bob"`.
8. End the program.

CONCLUSION:

This assignment provided an overview of Python basics, data types, control structures, and functions. Through the implementation and examples, we built a strong foundation for understanding and applying these essential programming concepts.