INTRODUCTION TO PYTHON PROGRAMMING

PYTHON :

Python is a popular, easy-to-learn programming language known for its simplicity and readability. It's used to write software programs for tasks like:

- Building websites (web development)
- Creating games (game development)
- Performing calculations (mathematics)

Python's syntax (rules for writing code) is straightforward, making it a great choice for beginners. For example, instead of using complicated symbols, Python uses easy words like print to display messages, and if to make decisions in code.

EXAMPLE :

Here's a very simple Python example:

```
print("Hello, World!")
```

OUTPUT: Hello, World!

COMMENTS IN PYTHON :

Single Line Comments:

```
# This is a comment. It does not affect the program.
```

Multi Line Comments:

```
"""This is a multiline comment.
It can span multiple lines.
You can use it to explain the code in detail.'''
```

Basic Syntax and Concepts :

- **Variables**: Storing data (e.g., name = "Alice")
- **Data Types**: Integers, floats, strings, lists, etc.
- **Comments**: Writing comments to explain code (# This is a comment)
- **Indentation**: Python uses indentation (spaces) to structure code (e.g., for loops, functions)

Operators in Python :

- Arithmetic operators (+, -, *, /)
- Comparison operators (==, !=, <, >)
- Logical operators (and, or, not)

Control Structures :

- **Conditionals**: if, else, elif statements to make decisions based on conditions
- **Loops**: for loop (iterating through a collection) and while loop (repeating as long as a condition is true)

Functions :

- Defining functions with def (e.g., def greet(name):)
- Returning values from functions

Working with Data :

- **Lists**: Storing collections of items (e.g., fruits = ["apple", "banana", "cherry"])
- **Dictionaries**: Storing key-value pairs (e.g., person = {"name": "Alice", "age": 25})
- **Tuples and Sets**: Immutable sequences and unique collections.

Error Handling :

- Using try, except blocks to handle errors (e.g., catching division by zero errors).

Modules and Libraries :

- Importing built-in libraries (e.g., import math to use math functions).
- Brief mention of popular libraries like NumPy (for math), Pandas (for data analysis), and Flask (for web development).

PYTHON PROGRAMS FOR PRACTICE

1. Hello World

- Print "Hello, World!" to the screen.

```
print("Hello, World!")
```

2. Sum of Two Numbers

- Ask the user for two numbers and print their sum.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
print("Sum:", num1 + num2)
```

3. Even or Odd

- Check if a number is even or odd.

```
num = int(input("Enter a number: "))
if num % 2 == 0:
   print("Even")
else:
   print("Odd")
```

4. Find the Largest Number

- Ask the user for three numbers and print the largest.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))
largest = max(num1, num2, num3)
print("The largest number is:", largest)
```

5. Simple Calculator

- Create a simple calculator that adds, subtracts, multiplies, or divides based on user input.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
operation = input("Enter operation (+, -, *, /): ")

if operation == "+":
  print(num1 + num2)
elif operation == "-":
  print(num1 - num2)
elif operation == "*":
  print(num1 * num2)
elif operation == "/":
  print(num1 / num2)
else:
  print("Invalid operation")
```

6. Factorial of a Number

- Calculate the factorial of a number.

```
num = int(input("Enter a number: "))
factorial = 1
for i in range(1, num + 1):
  factorial *= i
print("Factorial:", factorial)
```

7. Check Prime Number

- Check if a number is prime.

```
num = int(input("Enter a number: "))
is_prime = True
for i in range(2, num):
    if num % i == 0:
        is_prime = False
        break
if is_prime:
    print(num, "is prime.")
else:
    print(num, "is not prime.")
```

8. Fibonacci Sequence

- Print the Fibonacci sequence up to n terms.

```
n = int(input("Enter number of terms: "))
a, b = 0, 1
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

9. Count Vowels and Consonants

- Count the vowels and consonants in a given string.

```
text = input("Enter a string: ")
vowels = "aeiouAEIOU"
vowels_count = 0
consonants_count = 0
for char in text:
    if char.isalpha():
        if char in vowels:
            vowels_count += 1
        else:
            consonants_count += 1
print(f"Vowels: {vowels_count}, Consonants: {consonants_count}")
```

10. Reverse a String

- Reverse a string.

```
text = input("Enter a string: ")
print("Reversed string:", text[::-1])
```

11. Multiplication Table

- Print the multiplication table of a number.

```
num = int(input("Enter a number: "))
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

12. Palindrome Checker

- Check if a string is a palindrome.

```
text = input("Enter a string: ")
if text == text[::-1]:
    print("Palindrome")
else:
    print("Not a palindrome")
```

13. Armstrong Number

- Check if a number is an Armstrong number (sum of cubes of digits equals the number).

```
num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10
if sum == num:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

14. Print Prime Numbers in a Range

- Print all prime numbers in a given range.

```
start = int(input("Enter start of range: "))
end = int(input("Enter end of range: "))
for num in range(start, end + 1):
    is_prime = True
    for i in range(2, num):
        if num % i == 0:
            is_prime = False
            break
    if is_prime:
        print(num, end=" ")
```

15. Sum of Digits

- Find the sum of digits of a number.

```
num = int(input("Enter a number: "))
sum_digits = 0
while num > 0:
    sum_digits += num % 10
    num //= 10
print("Sum of digits:", sum_digits)
```

16. Find GCD of Two Numbers

- Find the greatest common divisor (GCD) of two numbers.

```
def gcd(x, y):
    while(y):
        x, y = y, x % y
    return x
```

```
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print("GCD:", gcd(num1, num2))
```

17. Count Words in a String

- Count the number of words in a string.

```
text = input("Enter a string: ")
words = text.split()
print("Word count:", len(words))
```

18. Create a Dictionary from Two Lists

- Create a dictionary from two lists (keys and values).

```
keys = ['name', 'age', 'city']
values = ['Alice', 25, 'New York']
my_dict = dict(zip(keys, values))
print(my_dict)
```

19. Print a Pyramid Pattern

- Print a simple pyramid pattern.

```
rows = int(input("Enter number of rows: "))
for i in range(1, rows + 1):
    print(" " * (rows - i) + "*" * (2 * i - 1))
```

20. Find LCM of Two Numbers

- Find the least common multiple (LCM) of two numbers.

```
def lcm(x, y):
    greater = max(x, y)
    while True:
        if greater % x == 0 and greater % y == 0:
            return greater
        greater += 1
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print("LCM:", lcm(num1, num2))
```