Questions

1. Define a variable **age** with your age and print it.
2. Create a variable with a valid identifier for your favourite color and assign it a value. Print the variable.
3. Swap the values of two variables **a** and **b** using a temporary variable.
4. Write a program that performs all arithmetic operations between two numbers **x** and **y**.
5. Compare two variables **m** and **n** and print whether **m** is greater than **n**.
6. Write a program using logical operators to check if a number is between 10 and 20.
7. Define a string variable and print its length.
8. Create a list of five numbers and print the sum of the list.
9. Create a dictionary with keys as names and values as ages. Print the age of a specific name.
10. Write a program that checks if a number is even or odd.
11. Write a program to check if a number is positive, negative, or zero using nested if.
12. Write a program that assigns a grade based on a score.
13. Print all even numbers from 1 to 20 using a for loop.
14. Print numbers from 10 down to 1 using a while loop.
15. Print a multiplication table from 1 to 5 using nested loops.
16. Write a function **greet** that prints "Hello, World!".
17. Write a function that takes two numbers and returns their sum.
18. Write a function that returns the factorial of a number.
19. Write a function that greets a person with a default name if no name is provided.
20. Write a function that takes a variable number of arguments and prints each one.
21. Write a program that prints the first 10 numbers in the Fibonacci sequence.
22. Create a function called **add\_numbers** that takes two numbers as parameters and returns their sum.
23. Python function **is\_even** that takes an integer and returns **True** if the number is even and **False** otherwise.
24. Implement a function **is\_prime** that checks if a given number is a prime number.
25. Write a function **sum\_list** that takes a list of numbers and returns the sum of all the numbers in the list. List=[1,2,3,4,5]

age = 25

print(age)

favorite\_color = "blue"

print(favorite\_color)

a = 3

b = 8

temp = a

a = b

b = temp

print("a:", a)

print("b:", b)

x = 12

y = 4

print("Addition:", x + y)

print("Subtraction:", x - y)

print("Multiplication:", x \* y)

print("Division:", x / y)

print("Modulus:", x % y)

m = 10

n = 5

print(m > n)

num = 15

print(10 < num < 20)

message = "Hello, Python!"

print(len(message))

numbers = [1, 2, 3, 4, 5]

print(sum(numbers))

ages = {"Alice": 25, "Bob": 30, "Charlie": 35}

print(ages["Bob"])

num = 7

if num % 2 == 0:

print("Even")

else:

print("Odd")

num = -10

if num >= 0:

if num == 0:

print("Zero")

else:

print("Positive")

else:

print("Negative")

score = 85

if score >= 90:

print("A")

elif score >= 80:

print("B")

elif score >= 70:

print("C")

elif score >= 60:

print("D")

else:

print("F")

for i in range(1, 21):

if i % 2 == 0:

print(i)

i = 10

while i > 0:

print(i)

i -= 1

for i in range(1, 6):

for j in range(1, 6):

print(i \* j, end=' ')

print()

def greet():

print("Hello, World!")

greet()

def add(a, b):

return a + b

result = add(3, 5)

print(result)

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n - 1)

print(factorial(5))

def greet(name="Stranger"):

print("Hello,", name)

greet("Alice")

greet()

def print\_all(\*args):

for arg in args:

print(arg)

print\_all(1, 2, 3, "apple", "banana")

# Control Loops

a, b = 0, 1

for \_ in range(10):

print(a, end=' ')

a, b = b, a + b

# Output should be 0 1 1 2 3 5 8 13 21 34

def add\_numbers(a, b):

return a + b

# Example usage:

result = add\_numbers(5, 3)

print(result) # Output: 8

def is\_even(number):

return number % 2 == 0

# Example usage:

print(is\_even(4)) # Output: True

print(is\_even(7)) # Output: False

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

# Example usage:

print(is\_prime(11)) # Output: True

print(is\_prime(4)) # Output: False

def sum\_list(numbers):

total = 0

for number in numbers:

total += number

return total

# Example usage:

print(sum\_list([1, 2, 3, 4, 5])) # Output: 15

**Questions On Python Basic & Advance**

1.Implement a Python script that checks if two strings are anagrams of each other (i.e., contain the same characters with the same frequency).

2.Create a Python script that uses variables to simulate a simple calculator. The program should handle addition, subtraction, multiplication, and division with input from the user.

3.Write a Python function that takes a list of integers and returns a new list with each element squared, but only if it is greater than 10.

4.Implement a program that checks if a given number is a power of two using bitwise operators.

5.Write a Python function that merges two dictionaries by adding values for common keys. If the key is present in both dictionaries, add their values together.

6.Write a Python program that takes three integers as input and prints them in non-decreasing order without using built-in sorting methods.

7.Write a Python function that takes a positive integer n and prints the Fibonacci sequence up to the nth term using a for loop.

8.Write a Python function that performs division of two numbers. If the denominator is zero, catch the exception and print an error message.

9.Write a Python program that extracts all dates in the format "DD-MM-YYYY" from a given text string using regular expressions.

text = "John was born on 12-05-1995 and Mary on 30-08-1999."

10.Write a Python class Employee with attributes name, salary, and department. Implement a method to calculate the yearly salary and a class method to count the total number of employees created.

11.What is the difference between is and == operators in Python? Provide an example where using one instead of the other can lead to unexpected results.

12.What is the difference between \*args and \*\*kwargs in Python functions? Provide an example where both are used in a single function.

13.What is the purpose of the else block in Python’s exception handling? How does it differ from the finally block, and when would you use each?

14.Explain how you can write a decorator that takes arguments.

15.How to Create, Read, and Write to a Text File in Python (with Examples)