ASSESSMENT 2:

1. What is a string in Python?

A string is a sequence of characters enclosed within quotes

X=”Hello”

2. How do you declare a string literal in Python?

By enclosing text within single ('), double ("), or triple quotes (''' or """).

s1 = 'Hello'

s2 = "World"

s3 = '''Multiline

String'''

3. Which operator is used to concatenate two strings?

Concatenation means joining strings.

a = "Hello"

b = "World"

print(a + b)

4. How do you access the first character of a string `s`?

Strings are indexed from 0, so the first character is at index 0.

s = "Python"

print(s[0])

5. What will be the output of `len('Hello')`?

The len() function returns the number of characters in a string.

Print(len('Hello'))

6. Which method is used to convert all characters of a string to uppercase?

upper()**:**This method returns a new string with all characters in uppercase.

s = "python"

print(s.upper())

7. How do you check if a string `s` starts with the letter `'A'`?

s.startswith('A'): The startswith() method returns True if the string begins with the specified substring.

s = "Apple"

print(s.startswith('A'))

8. What does the `strip()` method do in Python?

It removes leading and trailing whitespace

s = " hello "

print(s.strip())

9. What is the difference between `isalpha()` and `isdigit()` string methods?

* salpha() checks if all characters are alphabetic (A–Z or a–z).
* isdigit() checks if all characters are digits (0–9).

10. How can you replace all occurrences of the letter `'a'` with `'@'` in a string `s`?

The replace() method substitutes all matching characters.

s = "banana"

print(s.replace('a', '@'))

…………………………………………………………………………………………………………………………………………

11. Write the syntax and usage of the `find()` method.

Syntax: string.find(substring, start, end)  
Usage: Returns the lowest index of the substring if found, else returns -1.

s = "Hello World"

print(s.find("o"))

print(s.find("z"))

12. What is the difference between the `find()` and `index()` methods?

Both return the index of the first occurrence of a substring.

* find() returns -1 if the substring is not found.
* index() raises a ValueError if the substring is not found.

13. What is the return type of the `split()` method?

A list of strings.

s = "a,b,c"

print(s.split(','))

14. How do you join a list of strings into a single string?

Using join() method.

words = ['Hello', 'World']

print(' '.join(words))

15. Which method checks if all characters in a string are lowercase?

islower()

"hello".islower() # True

"Hello".islower() # False

16. How do you convert a string to title case in Python?

Using title() method

s = "hello world"

print(s.title()) # Output: Hello World

17. What will be the output of `'Hello'.lower()`?

hello will be the output

18. How can you count the number of occurrences of a substring in a string?

Using count() method

s = "banana"

print(s.count('a'))

19. Write an example demonstrating the use of `startswith()` method.

s = "Shreya gore"

print(s.startswith("shreya"))

print(s.startswith("J"))

20. What will be the output of `' Hello '.lstrip()`?

lstrip() removes leading spaces

print(' Hello '.lstrip())

…………………………………………………………………………………………………………………………………………………..

21. What is the difference between `=` and `==` operators in Python?

= is the assignment operator: It assigns a value to a variable.

== is the equality comparison operator: It checks if two values are equal.

22. What is the meaning of `a += 5` in Python?

a+=5 has the meaning of a = a + 5.

Example:

a = 10

a += 5

print(a)

23. What happens when you write `a = b = 10` in Python?

Both a and b are assigned the same value 10.

24. Explain the working of `a \*= 2` with an example.

It multiplies the value of a by 2 and assigns the result back to a. Equivalent to:  
a = a \* 2

Example;

a = 4

a \*= 2

print(a)

25. Which operator is used for floor division assignment in Python?

//= is used for floor division

Example:

a = 10

a //= 3

print(a) (because 10 // 3 = 3)

26. List all arithmetic operators used in Python.

+ → Addition

- → Subtraction

\* → Multiplication

/ → Division (floating-point)

// → Floor Division

% → Modulus

\*\* → Exponentiation (Power)

27. What will be the output of `5 + 3 \* 2` in Python?

In operator precedence, multiplication (\*) is evaluated before addition (+):  
3 \* 2 = 6, then 5 + 6 = 11

28. What will be the result of `10 / 3` in Python?

3.3333333333333335. It gives float.

29. What is the difference between `/` and `//` operators?

/ returns floating-point division (e.g., 10 / 3 = 3.33)

// returns floor division (e.g., 10 // 3 = 3)

30. What will be the result of `2 \*\* 3` in Python?

8 is the answer (2\*2\*2=8)

31. What is the modulus operator used for?

To find the **remainder** after division.

32. What will be the output of `-5 % 3` in Python?

33. What is the precedence order of arithmetic operators in Python?

1 \*\* (Exponentiation)

2 \*, /, //, % (Multiplication, Division, Floor Division, Modulus)

3 +, - (Addition, Subtraction)

34. What will be the result of `4 + 3 - 2 \* 2 / 1`?

1. 2 \* 2 = 4
2. 4 / 1 = 4.0
3. 4 + 3 = 7, then 7 - 4.0 = 3.0

35. Which arithmetic operator is used to calculate powers in Python?

\*\* (Exponentiation)

Ex: 2 \*\* 4 = 16

…………………………………………………………………………………………………………………………………………………

36. Write a Python expression that concatenates `'Hello'` and `'World'` with a space in between.

Expression- 'Hello' + ' ' + 'World'

37. Given `s = "Hello"`, write Python code to print the last character of `s`.

print(s[-1])

38. If `a = 5`, write an expression to multiply `a` by `2` using the assignment operator.

a \*= 2

39. Given `s = "Python"`, write Python code to print the substring `'yth'`.

print(s[1:4])

40. Write a Python expression to check if the word `'apple'` is present in the string `'I have an

apple'`.

S='I have an apple'

Print(S.find('apple'))