Section 1: Strings - Basic Concepts

1.) What is a string in Python?

A string is a sequence of characters enclosed in quotes.

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

We can declare a string by writing text in single quotes like 'hello' or double quotes like "hello".

3.) Which operator is used to concatenate two strings?

The + operator is used to concatenate two strings.

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

To get the first character of string s, we use s[0].

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

len('Hello') will give 5 because there are 5 characters.

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

.upper()

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

s.startswith('A')

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

The strip() method removes any spaces or newlines from the beginning and end of a string.

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

=> isalpha() checks if all characters are letters =>isdigit() checks if all characters are digits.

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

To replace 'a' with '@' in s, we use s.replace('a', '@').

Section 2: String Methods - Usage & Examples

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

s.find('x') looks for 'x' in string s and gives its index. If it’s not there, it returns -1.

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

find() and index() are similar, but index() gives an error if the substring isn’t found, while find() just returns -1.

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

The split() method returns a list of strings

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

' '.join(list_of_strings)

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

s.islower()

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

s.title()

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

'hello'

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

s.count(substring)

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

'Mango'.startswith('M') will return True

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

'Hello '

Section 3: Assignment Operators - Conceptual & Examples

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

= is for assignment.

== is for comparison.

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

a += 5 means add 5 to a and store the result back in a

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

Both a and b are assigned the value 10.

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

a *= 2 multiplies ‘a’ by 2 and saves it again in ‘a’. For example, if a = 4, then after a *= 2, a becomes 8

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

//=

Section 4: Arithmetic Operators - Concepts & Examples

26.) List all arithmetic operators used in Python.

The arithmetic operators are + (add), - (subtract), * (multiply), / (divide), // (floor divide), % (modulo), and ** (power).

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

5 + 3 * 2 will give 11 because multiplication happens first (3*2 = 6, then 5+6 = 11)

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

3.3333333333333335

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

/ gives float division, // gives floor (integer) division.

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

2 ** 3 gives 8 because it's 2 to the power 3.

31.) What is the modulus operator used for?

The % operator gives the remainder after division.

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

1

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

** > *, /, //, % > +, -

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

4 + 3 - 2 * 2 / 1 will give 3.0

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

**

Section 5: Mixed Code-based Questions

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

'Hello' + ' ' + 'World' will give '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 will give a = 10.

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'`.

'apple' in 'I have an apple' → True