**ASSIGNMENT 12**

Q1. Does assigning a value to a string’s indexed character violate Python’s string immutability?

Ans: We cannot assign any value to a string’s indexed character; therefore, it doesn’t violate Python’s string immutability.

Q2. Does using the += operator to concatenate strings violate Python’s string immutability? Why or why not?

Ans: No, using +=operator doesn’t violate Python’s string immutability because it is not changing the string but adding the new value to the existing string variable.

Q3. In Python, how many different ways are there to index a character?

Ans: There are two different ways to index a character. First is positive indexing which starts from 0 to length of the string and second is negative indexing I.e., starting from end using –1 index. For example, if we have a string string=’Hello’, then we can access ‘o’ by string[4] or string[-1].

Q4. What is the relationship between indexing and slicing?

Ans: Slicing is the indexing of a string to create a substring in the form of str[a:b], where a is the starting index of str and b is not included ending index of str. For example, string=’Hello’ str[2:4] will give ‘ll’ output.

Q5. What is an indexed character’s exact data type? What is the data form of a slicing-generated

substring?

Ans: Data type of indexed character and of slicing-generated substring is string.

Q6. What is the relationship between string and character “types” in Python?

Ans: Python doesn’t have character types but a character is considered as a string.

Q7. Identify at least two operators and one method that allow you to combine one or more smaller

strings to create a larger string.

Ans: To combine two or more strings we can use ‘+’ or ‘+=’ operator in python. For example, if we have s=’a’ and v=’y’ then z=s+y will give ‘ay’ and then if we use z+=s it will give ‘aya’. The same result we can achieve by the join method by using the syntax ““.join([s,v]), where we can have any value in the quotes to attach the string to and s, v are the strings.

Q8. What is the benefit of first checking the target string with in or not in before using the index

method to find a substring?

Ans: It is useful because if the substring is not present in the string, then the index method will give error and stop the program.

Q9. Which operators and built-in string methods produce simple Boolean (true/false) results?

Ans: Operators like and or and not produce Boolean results and string methods like str.islower(), str.isalpha(), str.isdigit() and many more return Boolean results.