Assignment-6

1.What are escape characters, and how do you use them?

Ans1- Escape characters in programming are special characters that are used to represent characters that are difficult or impossible to type directly as part of a string or character literal. They allow you to include characters in a string that would otherwise have a special meaning or be interpreted differently by the programming language. Escape characters are usually represented by a backslash **\** followed by another character.

Examples-\\ ,\n , \t etc

2. What do the escape characters n and t stand for?

Ans2- \n : it stands for Newline , and it is used to move to new line .

\t: it stands for Tab , and it is used for adding horizontal spacing.

3. What is the way to include backslash characters in a string?

Ans3-To include a backslash character (**\**) in a string in Python, you need to use a double backslash **\\**. This is because the backslash is an escape character in Python, so using a single backslash would be interpreted as the start of an escape sequence. To represent a literal backslash, you need to escape it with another backslash.

4. The string "Howl's Moving Castle" is a correct value. Why isn't the single quote character in the word Howl's not escaped a problem?

Ans4-In Python, you can include single quotes inside a string enclosed in double quotes, and vice versa, without any issue. This is because Python allows you to use either single quotes (**'**) or double quotes (**"**) to define string literals, and you can freely mix them as long as you properly match the opening and closing quotes.

5. How do you write a string of newlines if you don't want to use the n character?

Ans5-If you don't want to use the newline escape character **\n**, and you want to include newlines directly in a string, you can achieve this by using triple-quoted strings.

my\_string = """This is a string

that spans multiple lines

without using the '\\n' character."""

6. What are the values of the given expressions?

'Hello, world!'[1] = ‘e’

'Hello, world!'[0:5] = ‘Hello’

'Hello, world!'[:5] = ‘Hello’

'Hello, world!'[3:] = ‘lo , world!’

7. What are the values of the following expressions?

'Hello'.upper() = HELLO

'Hello'.upper().isupper() = returns True

'Hello'.upper().lower() = hello

8. What are the values of the following expressions?

'Remember, remember, the fifth of July.'.split() = ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']

'-'.join('There can only one.'.split()) = 'There-can-only-be-one.'

9. What are the methods for right-justifying, left-justifying, and centering a string?

Ans9 – Right-Justify:

Use the **str.rjust(width, fillchar)** method to right-justify a string within a specified **width**. You can also provide an optional **fillchar** parameter to specify the character used for padding.

Left-Justify:

Use the **str.ljust(width, fillchar)** method to left-justify a string within a specified **width**. Similar to **rjust()**, you can also provide an optional **fillchar** parameter for padding.

Center:

Use the **str.center(width, fillchar)** method to center a string within a specified **width**. Again, you can provide an optional **fillchar** parameter for padding.

10. What is the best way to remove whitespace characters from the start or end?

Ans10- To remove whitespace characters (such as spaces, tabs, and newline characters) from the start or end of a string in Python, you can use the **str.strip()** method. The **strip()** method removes leading (start) and trailing (end) whitespace by default