By default, the StreamWriter class overwrites the existing content in the file unless you explicitly tell it to append data. Here's a breakdown of what happens with each method:

**1. WriteLine(str):**

* This method writes a line (the string str) to the buffer associated with the StreamWriter.
* The data is not immediately written to the file; instead, it is temporarily stored in the buffer (in memory).

**2. Flush():**

* This method forces the buffered data to be written from memory to the file or output stream.
* If you don’t call Flush(), the buffered data may remain in memory and might not be written to the file until the StreamWriter is closed or disposed of.
* After calling Flush(), the buffer is cleared, but the stream remains open for further writing.

**3. Close():**

* This closes the StreamWriter and releases any associated resources.
* It also ensures that any remaining buffered data is written to the file (if Flush() hasn't already been called).
* Once the stream is closed, you can no longer write to it.

**Why is Existing Data Being Overwritten?**

By default, when you create a StreamWriter, it opens the file for writing in **overwrite mode**. This means that each time you write to the file, the existing content is removed, and the new content is written.

**Solution to Append Data:**

To append new data to the existing file content (instead of overwriting it), you need to open the StreamWriter in append mode by passing true as the second parameter to its constructor.

StreamWriter sw = new StreamWriter("file.txt", true)