BufferedWriter Class

Introduction

The **Java.io.BufferedWriter** class writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.Following are the important points about BufferedWriter −

- The buffer size may be specified, or the default size may be used.
- A Writer sends its output immediately to the underlying character or byte stream.

Class declaration

Following is the declaration for **Java.io.BufferedWriter** class −

```
public class BufferedWriter
   extends Writer
```

Field

Following are the fields for **Java.io.BufferedWriter** class −

- **protected Object lock** − This is the object used to synchronize operations on this stream.

Class constructors

Sr.No.	Constructor & Description
1	**BufferedWriter(Writer out)** This creates a buffered character-output stream that uses a default-sized output buffer.
2	**BufferedWriter(Writer out, int sz)** This creates a new buffered character-output stream that uses an output buffer of the given size.

Class methods

Sr.No.	Method & Description
1	void close() This method closes the stream, flushing it first.
2	void flush() This method flushes the stream.
3	void newLine() This method writes a line separator.
4	void write(char[] cbuf, int off, int len) This method writes a portion of an array of characters.
5	void write(int c) This method writes a single character.
6	void write(String s, int off, int len) This method writes a portion of a String.

Example of Java BufferedWriter

```
1.          package com.javatpoint;
2.          import java.io.*;
3.          public class BufferedWriterExample
4.           {
5.          public static void main(String[] args) throws Exception
6.           {
7.              FileWriter writer = new FileWriter("D:\\testout.txt");
8.              BufferedWriter buffer = new BufferedWriter(writer);
9.              buffer.write("Welcome to javaTpoint.");
10.             buffer.close();
11.             System.out.println("Success");
12.             }
13.         }
```