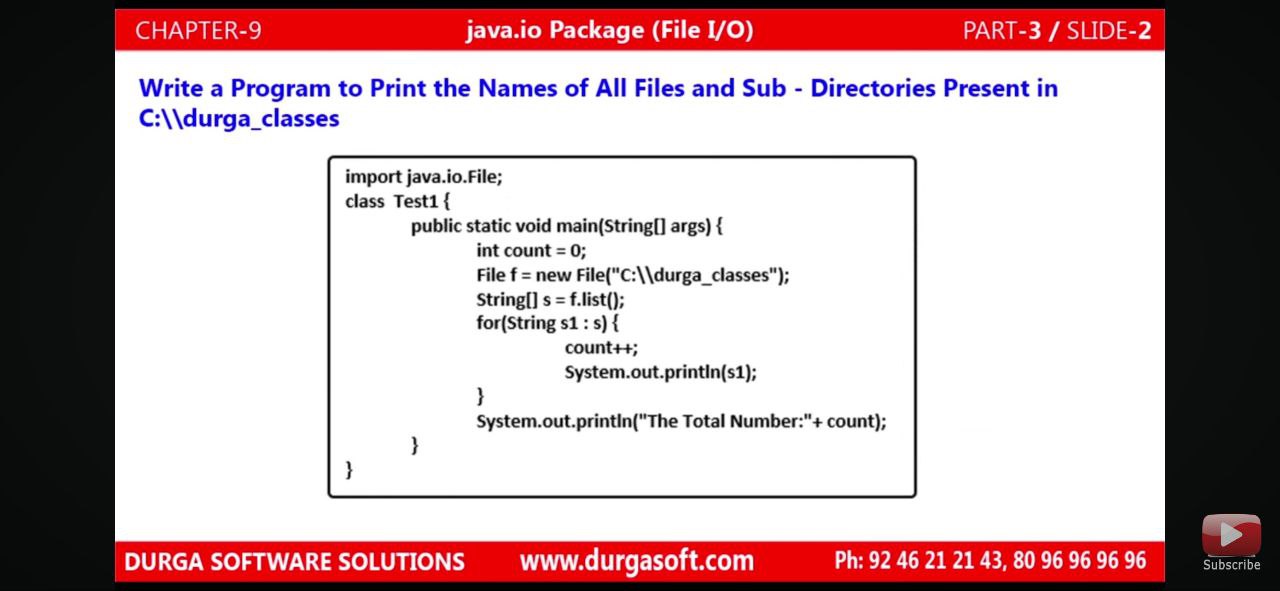
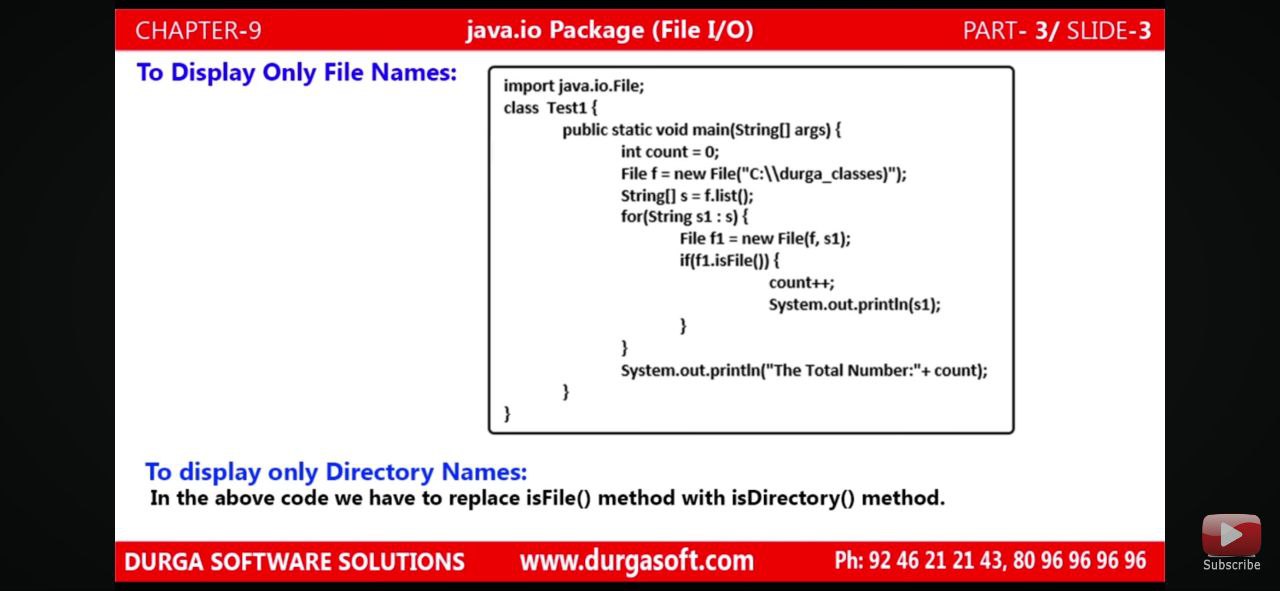
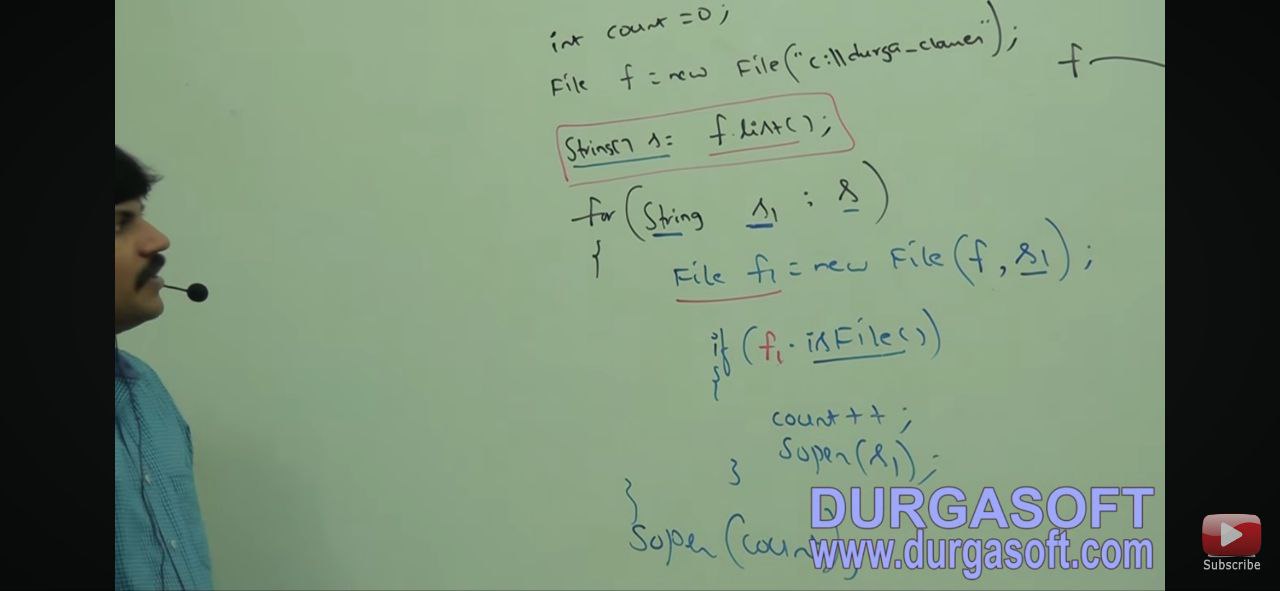
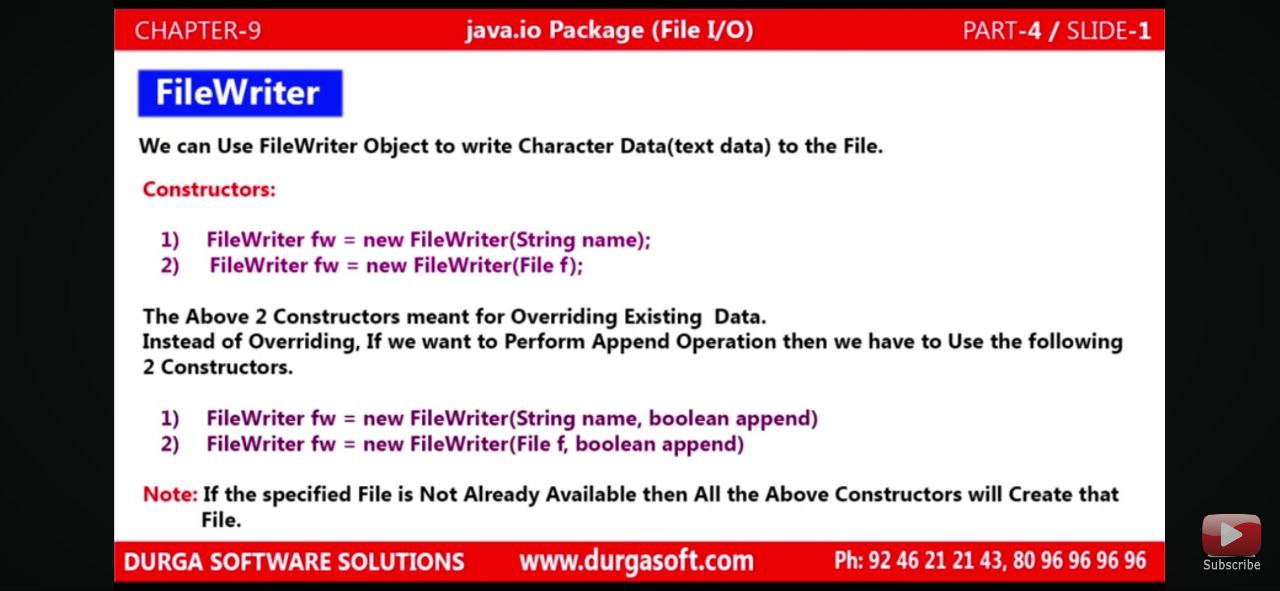


A red and white box with black text

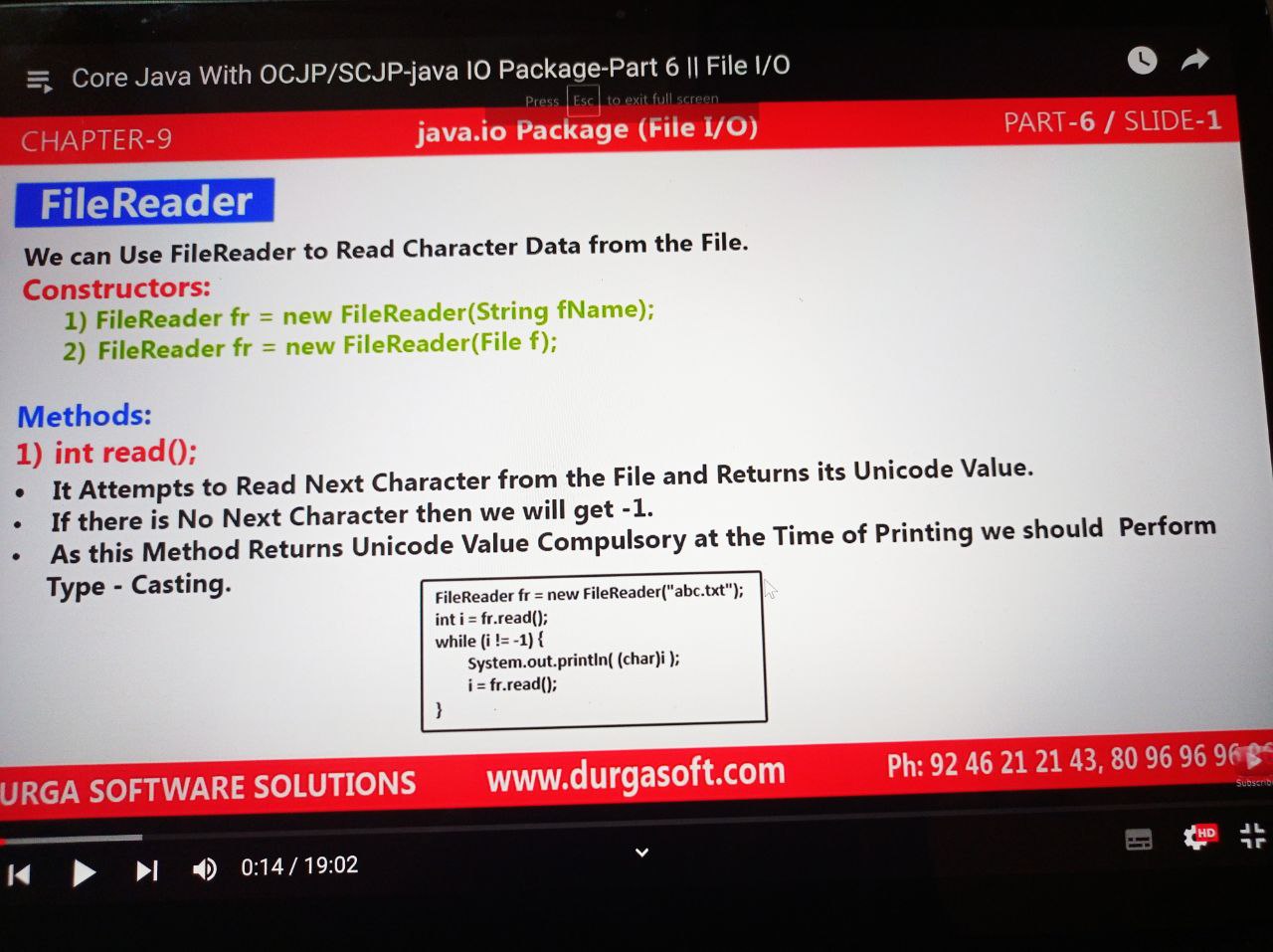
Description automatically generated





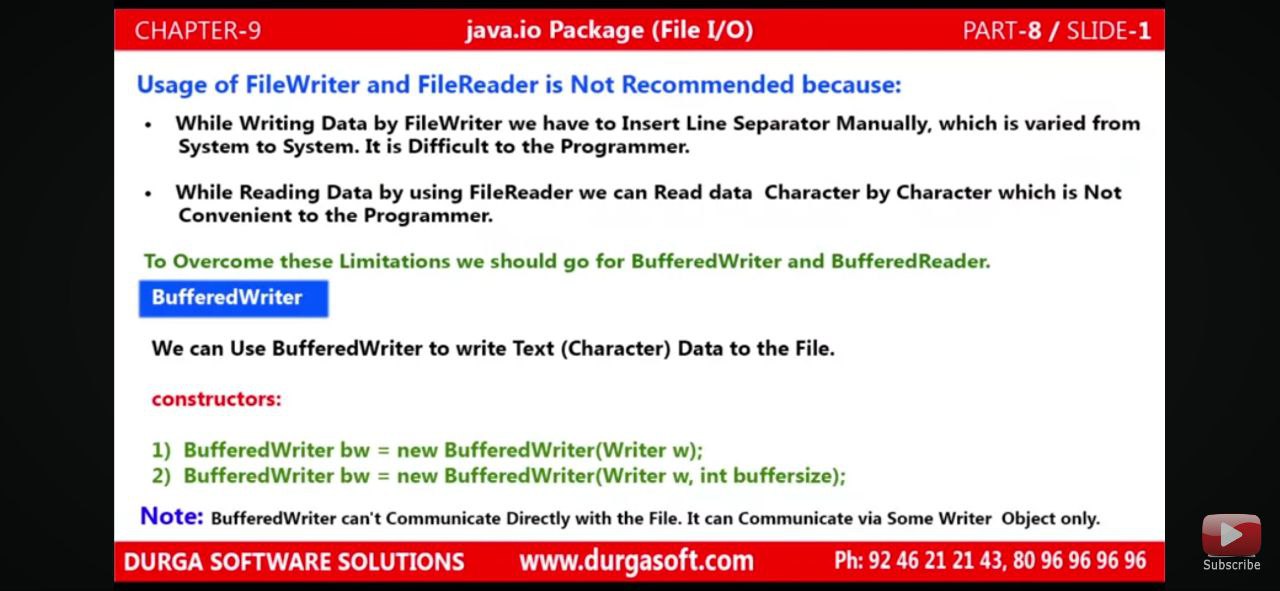


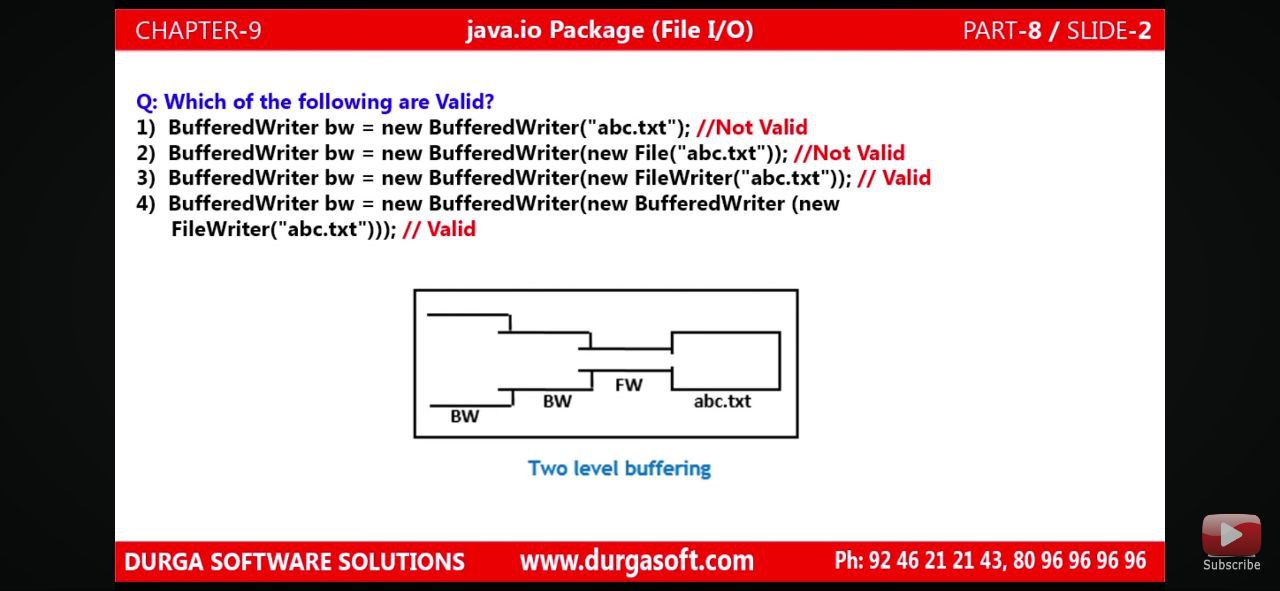
If File not exsit then it will create new file onward instead of using createNewFile() method here it can create File directly.



A screenshot of a computer

Description automatically generated

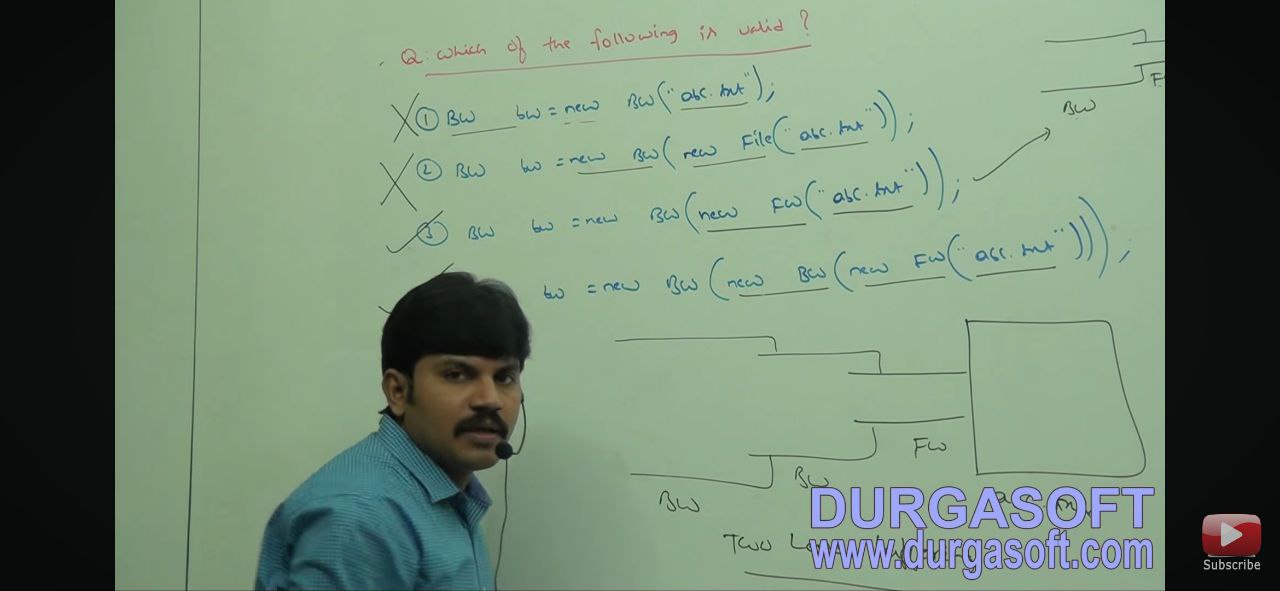


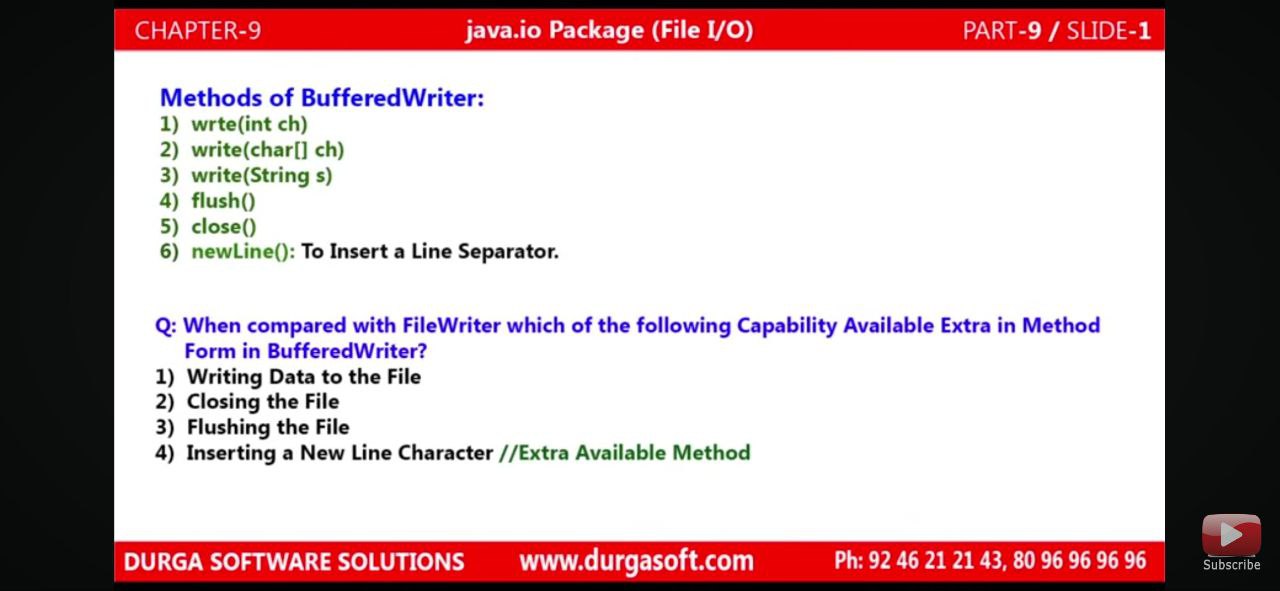


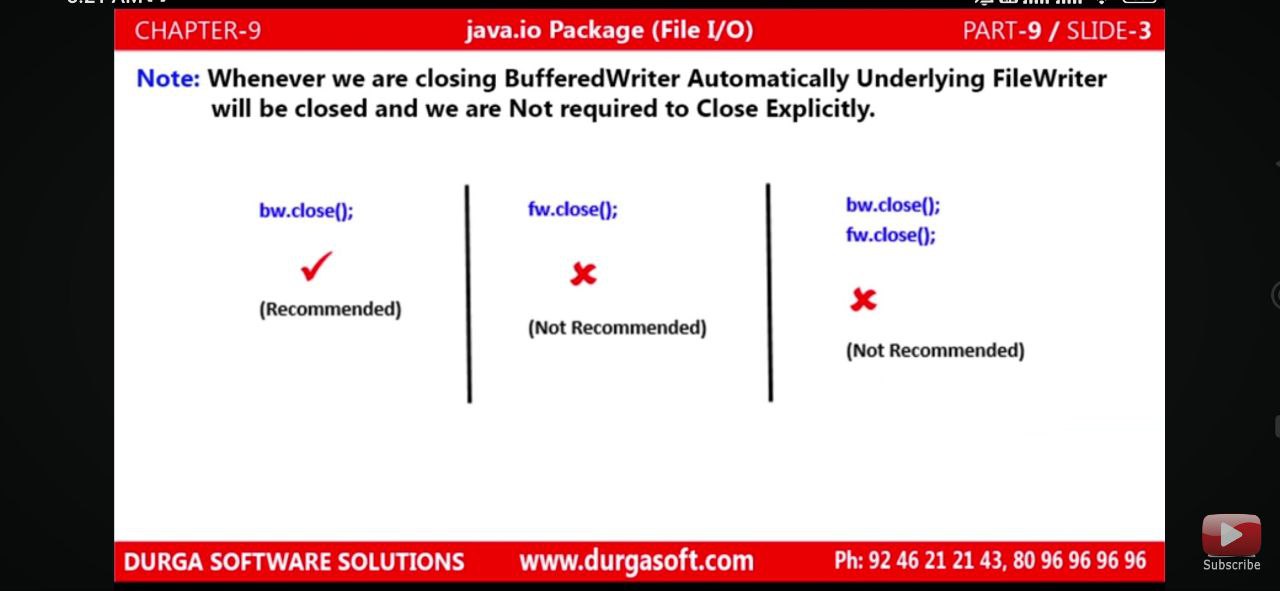
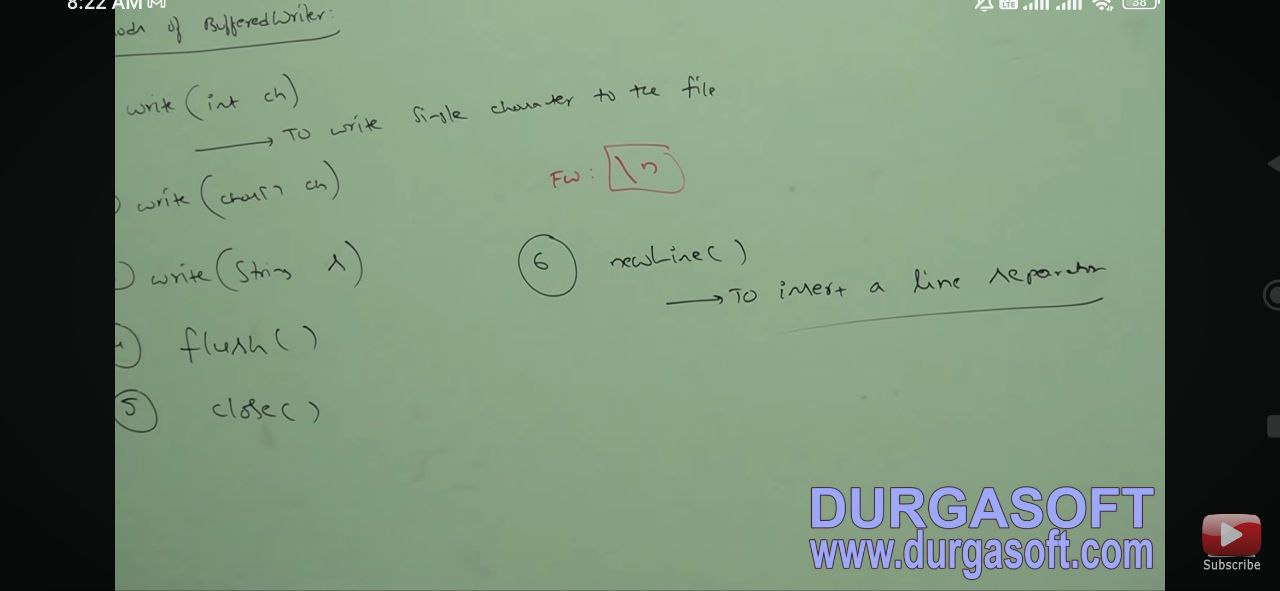
Because BufferWriter constructor not recommended to use File class. It’s only using writer classes.

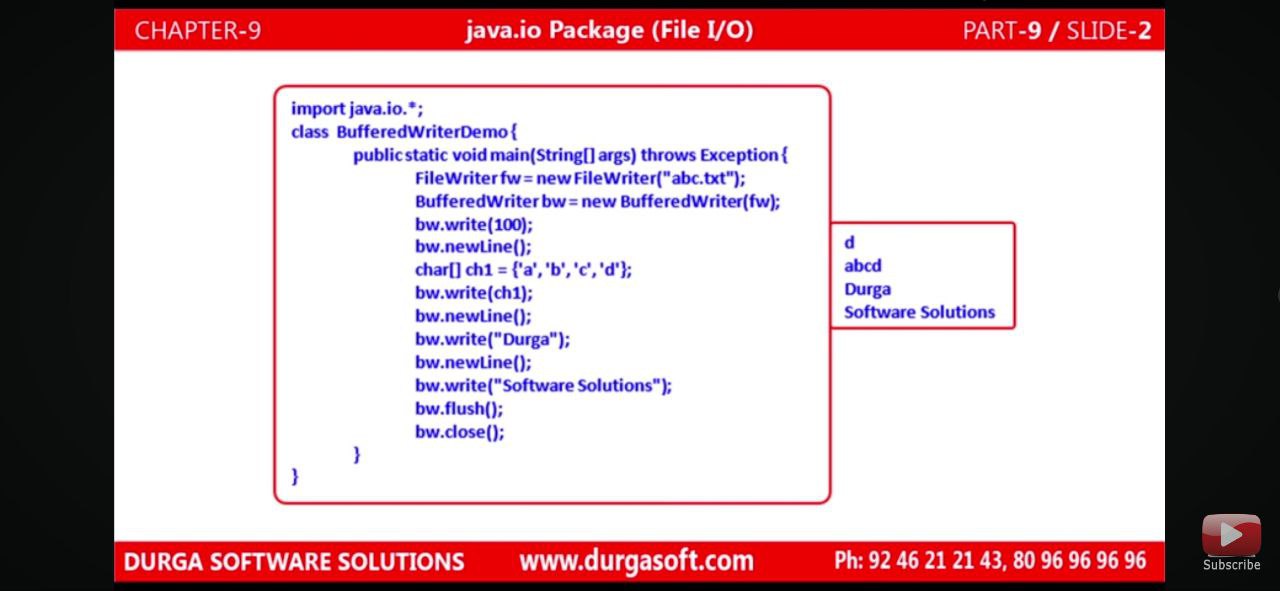
Why use BufferedWriter with FileWriter?

BufferedWriter: BufferedWriter is almost similar to FileWriter but it uses internal buffer to write data into File. So if the number of write operations is more, the actual IO operations are less and performance is better. You should use BufferedWriter when the number of write operations is more.





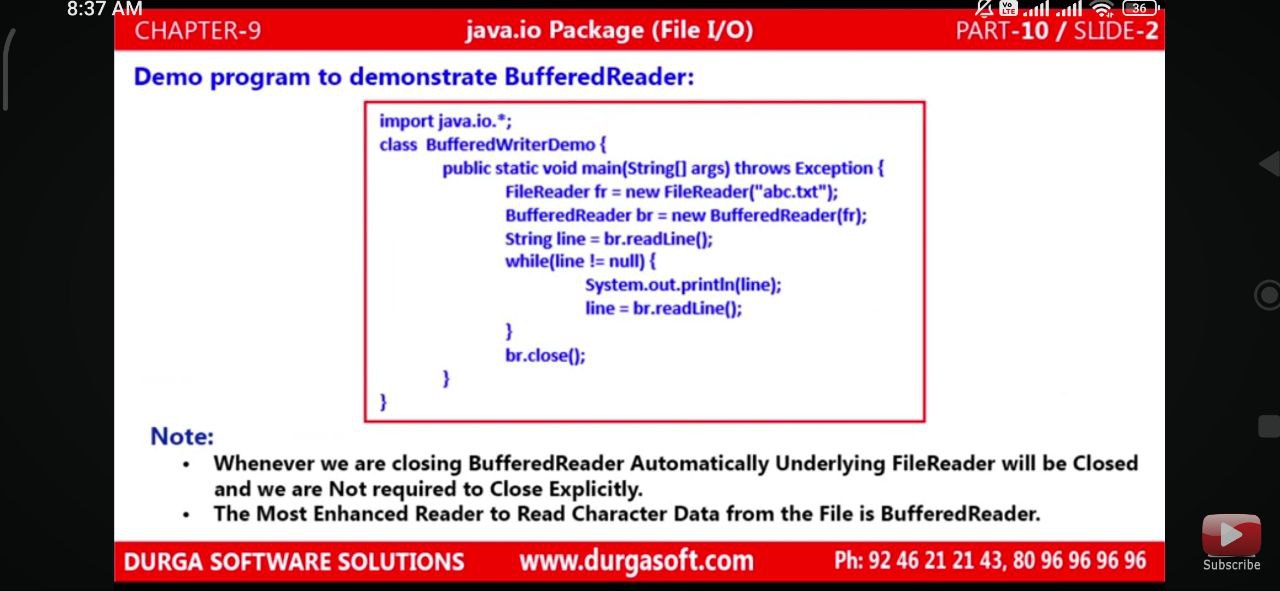




Problem we are facing with filereader is reading data character by character. that can solve by bufferreader which is reading character data line by line.



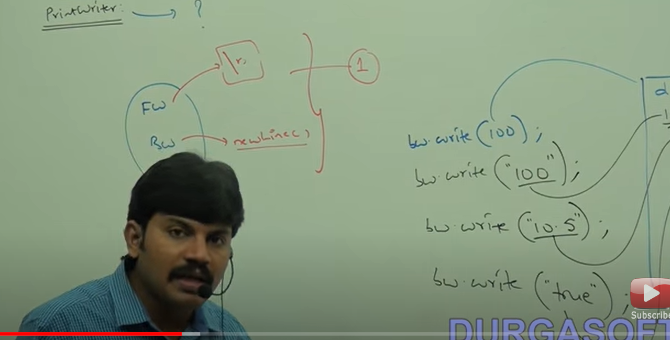
Note same with: Bufferwriter.



Cons of filewriter and bufferwriter: 1) In file-writer problem is to add(\n) while writing the character in line.

2)with bufferwriter problem is, if there is big code then we have to add Newline() every time in code that not good programming approach.

3)we can write only character data, if want to add any other type of data then compulsory we have to add inside (“ “)double quotes.

4) bufferwriter can’t communicate file directly. But printwriter can comminicate either with file object or directly.

To overcome this problem we can go with PrintWriter.

PrintWriter:

1)we can write any type of data in file directly.

Pw.print(100);

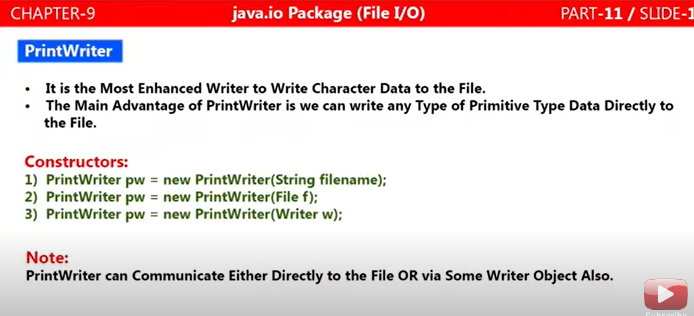
Pw.print(‘d’);

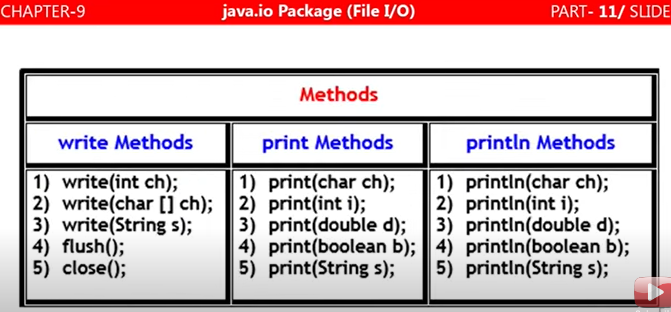
Pw.print(true);

Note: its not possible in filewriter and Bufferwriter

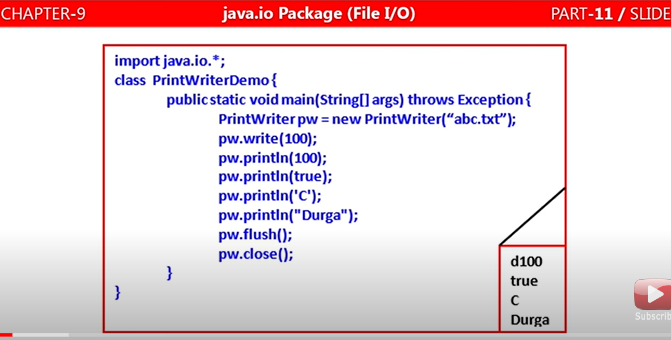
2)we can add (\n) directly.

Pw.println(100); //we can not required to call new line ( \n) separately.





Note : for line-separator we can directly use println method.

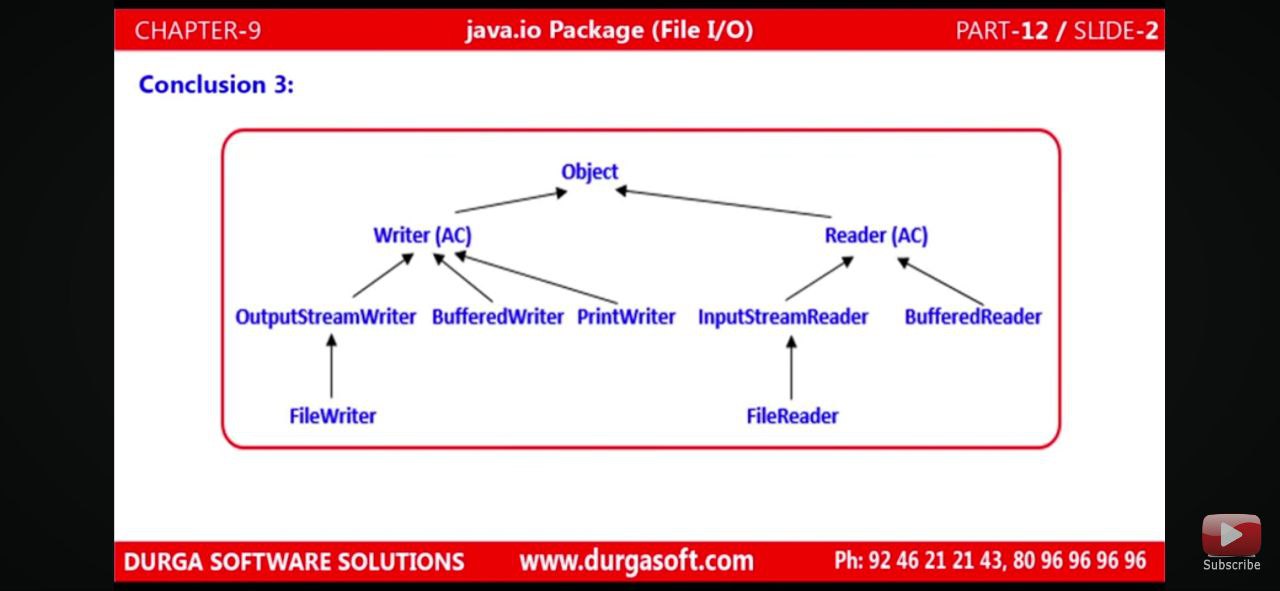


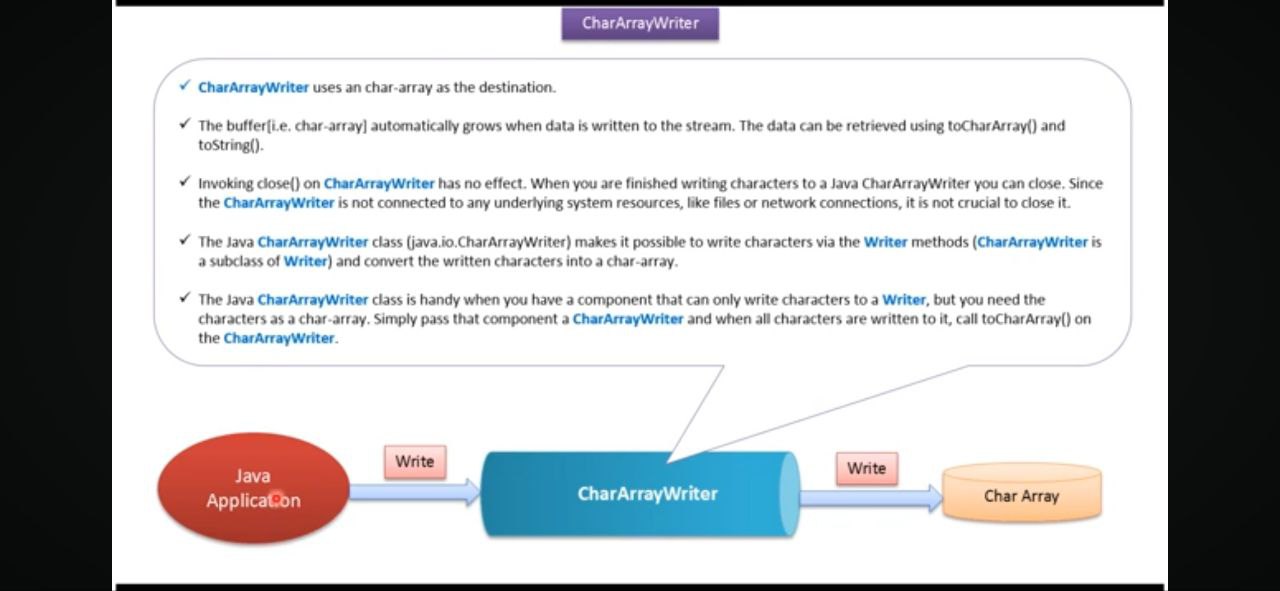
Note: If you are using write() method then it will print data in form of character only.

A screenshot of a computer

Description automatically generated







CharArrayWriter: is used to write the character Array data inside file. Its buffer automatically grows when data is written in this Class.

CharArrayReader: The CharArrayReader is composed of two words: CharArray and Reader. The CharArrayReader class is used to read character [array](https://www.javatpoint.com/array-in-java) as a reader. It inherits [Reader](https://www.javatpoint.com/java-reader-class) class.

CharArrayReader and CharArray writer: only reads data from Array and useful for small data only.

**CharArrayWriter and** CharArrayReader: Specifically designed for writing to character arrays.

**StringReader**

The **StringReader** class is used to create a character stream from a string. It allows you to treat a string as a character input stream, making it useful in situations where you need to read characters from a string using classes designed for character stream processing.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated