JAVA - DATAOUTPUTSTREAM

http://www.tutorialspoint.com/java/java_dataoutputstream.htm

The DataOutputStream stream let you write the primitives to an output source.

Following is the constructor to create a DataOutputStream.

```
DataOutputStream out = DataOutputStream(OutputStream  out);
```

Once you have *DataOutputStream* object in hand, then there is a list of helper methods, which can be used to write the stream or to do other operations on the stream.

SN	Methods with Description
1	**public final void write(byte[] w, int off, int len)throws IOException** Writes len bytes from the specified byte array starting at point off , to the underlying stream.
2	**Public final int write(byte [] b)throws IOException** Writes the current number of bytes written to this data output stream. Returns the total number of bytes write into the buffer.
3	**(a) public final void writeBooolean()throws IOException,** **(b) public final void writeByte()throws IOException,** **(c) public final void writeShort()throws IOException** **(d) public final void writeInt()throws IOException** These methods will write the specific primitive type data into the output stream as bytes.
4	**Public void flush()throws IOException** Flushes the data output stream.
5	**public final void writeBytes(String s) throws IOException** Writes out the string to the underlying output stream as a sequence of bytes. Each character in the string is written out, in sequence, by discarding its high eight bits.

Example:

Following is the example to demonstrate DataInputStream and DataInputStream. This example reads 5 lines given in a file test.txt and converts those lines into capital letters and finally copies them into another file test1.txt.

```
import java.io.*;

public class Test{
   public static void main(String args[])throws IOException{

      DataInputStream d = new DataInputStream(new
                               FileInputStream("test.txt"));

      DataOutputStream out = new DataOutputStream(new
                               FileOutputStream("test1.txt"));

      String count;
      while((count = d.readLine()) != null){
          String u = count.toUpperCase();
          System.out.println(u);
          out.writeBytes(u + "  ,");
      }
      d.close();
      out.close();
   }
}
```

Here is the sample run of the above program:

```
THIS IS TEST 1  ,
THIS IS TEST 2  ,
THIS IS TEST 3  ,
THIS IS TEST 4  ,
THIS IS TEST 5  ,
```