JAVA - DATAINPUTSTREAM

 Copyright © tutorialspoint.com

The DataInputStream is used in the context of DataOutputStream and can be used to read primitives.

Following is the constructor to create an InputStream:

```
InputStream in = DataInputStream(InputStream in);
```

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

SN	Methods with Description
1	**public final int read(byte[] r, int off, int len)throws IOException** Reads up to len bytes of data from the input stream into an array of bytes. Returns the total number of bytes read into the buffer otherwise -1 if it is end of file.
2	**Public final int read(byte [] b)throws IOException** Reads some bytes from the inputstream an stores in to the byte array. Returns the total number of bytes read into the buffer otherwise -1 if it is end of file.
3	**(a) public final Boolean readBooolean()throws IOException,** **(b) public final byte readByte()throws IOException,** **(c) public final short readShort()throws IOException** **(d) public final Int readInt()throws IOException** These methods will read the bytes from the contained InputStream. Returns the next two bytes of the InputStream as the specific primitive type.
4	**public String readLine() throws IOException** Reads the next line of text from the input stream. It reads successive bytes, converting each byte separately into a character, until it encounters a line terminator or end of file; the characters read are then returned as a String.

Example:

Following is the example to demonstrate DataInputStream and DataInputStream. This example reads 5 lines given in a file test.txt and convert 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  ,
```