Java ByteArrayOutputStream Class->

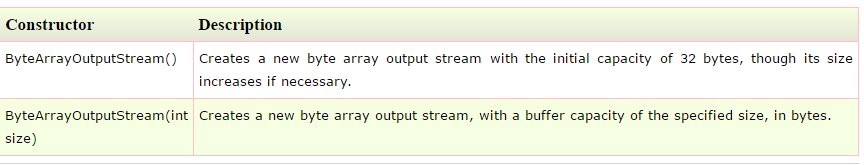
\*Java ByteArrayOutputStream class is used to **write common data** into multiple files.

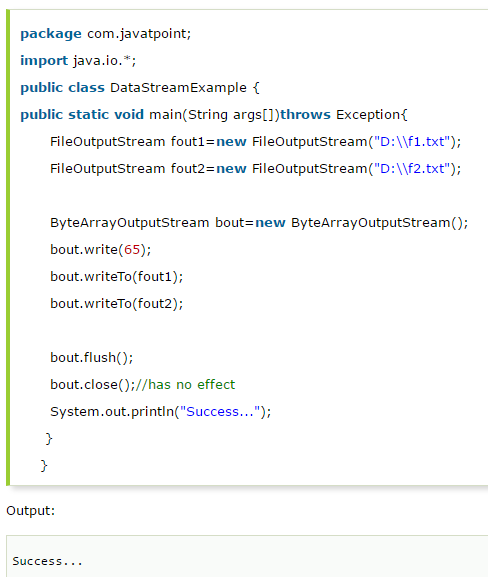
\*In this stream, the data is written into a byte array which can be written to multiple streams later.

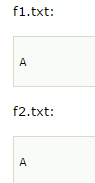
The ByteArrayOutputStream holds a copy of data and forwards it to multiple streams.

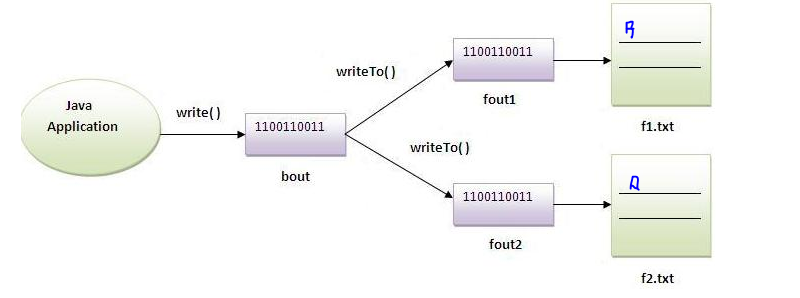
Declaration

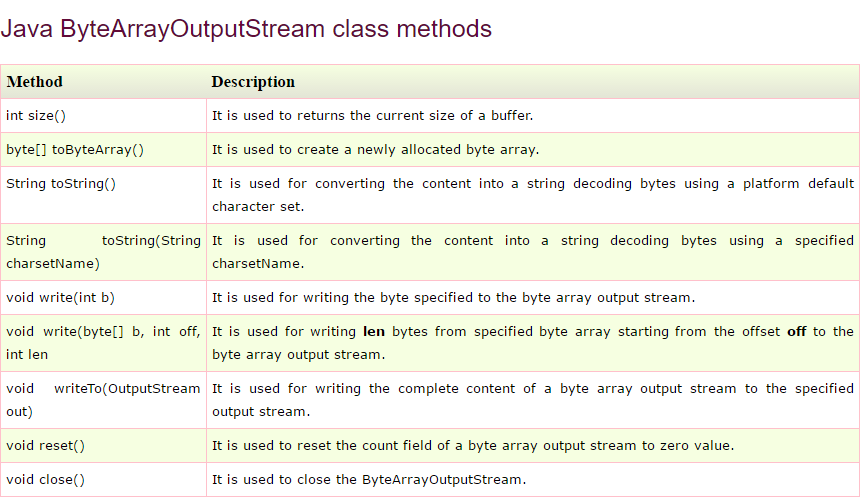
**public** **class** ByteArrayOutputStream **extends** OutputStream









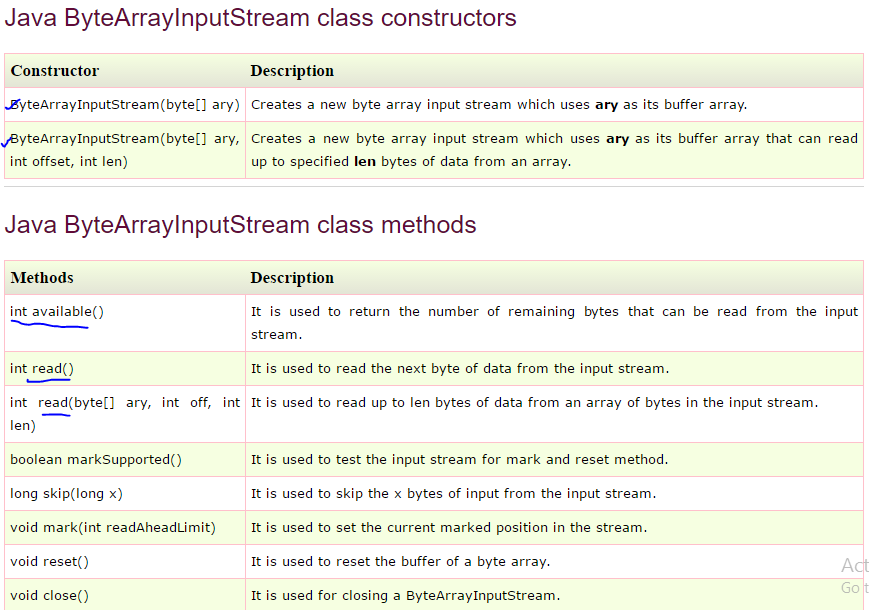


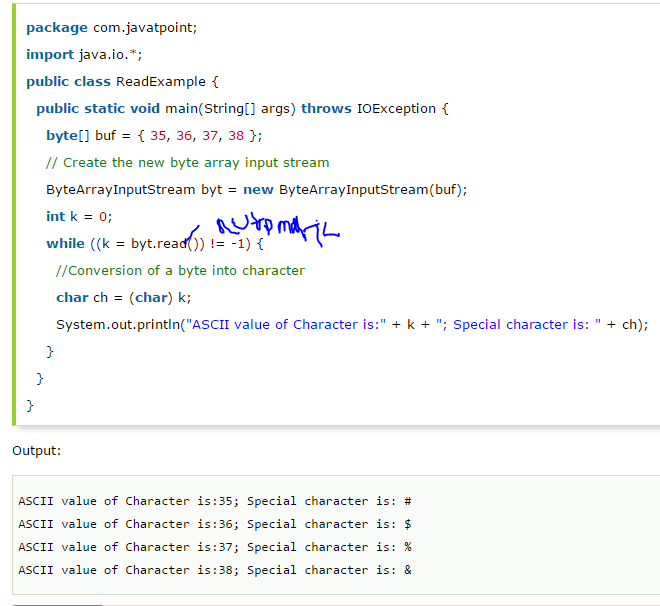
# Java ByteArrayInputStream Class->

# The ByteArrayInputStream is composed of two words: ByteArray and InputStream. As the name suggests, it can be used to read byte array as input stream.

# Java ByteArrayInputStream class contains an internal buffer which is used to ****read byte array**** as stream. In this stream, the data is read from a byte array.

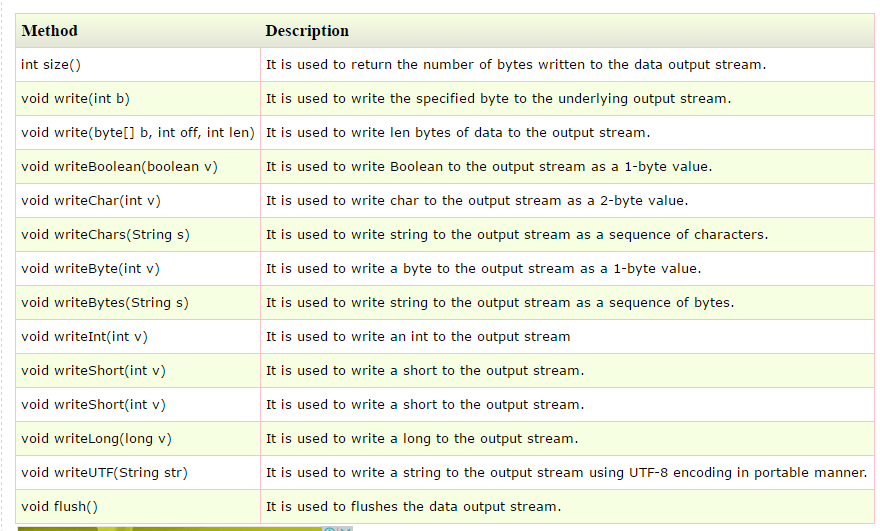
**public class ByteArrayInputStream extends InputStream**

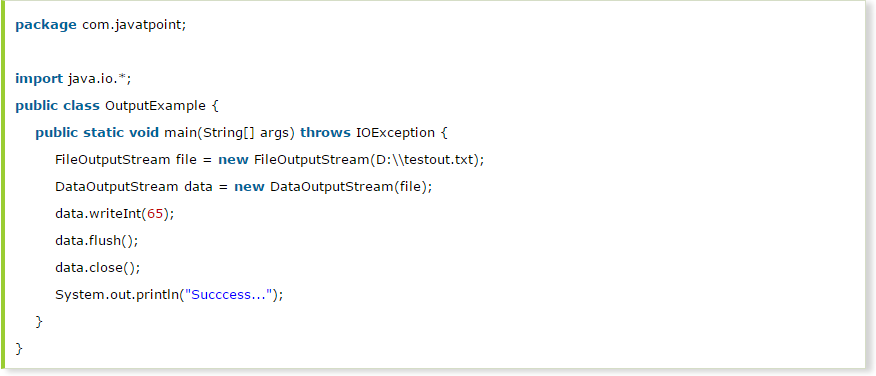




# Java DataOutputStream Class->

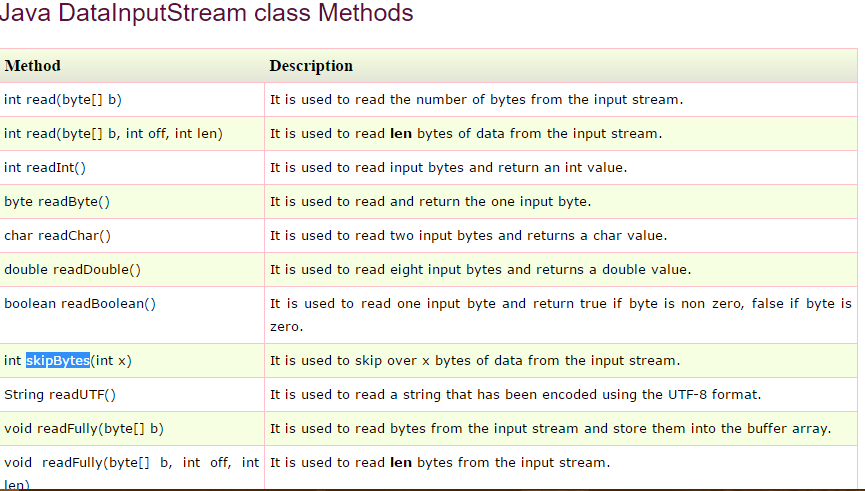
# Java DataOutputStream class allows an application to write primitive Java data types to the output stream in a machine-independent way.

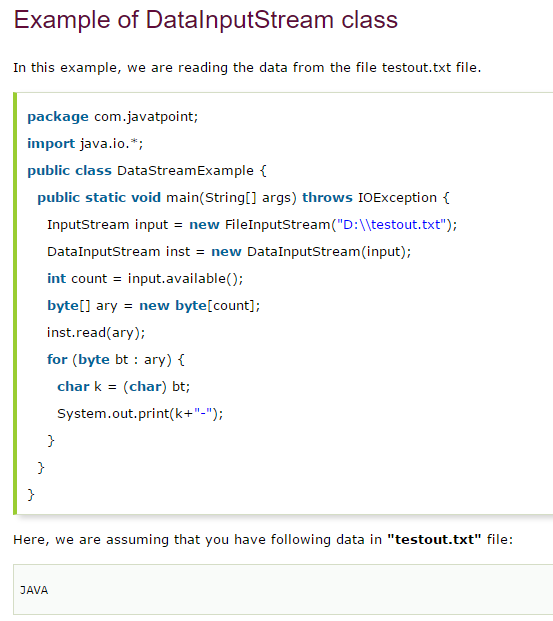




# Java DataInputStream Class->

# \*Java DataInputStream class allows an application to read primitive data from the input stream in a machine-independent way.





# Java FilterOutputStream Class->

# \*java FilterOutputStream class implements the OutputStream class.

# \*It provides different sub classes such as BufferedOutputStream and DataOutputStream to provide additional functionality.

# \* So it is less used individually.

# Java FilterInputStream Class->

# \*Java FilterInputStream class implements the InputStream.

# \*It contains different sub classes as BufferedInputStream, DataInputStream for providing additional functionality.

# \* So it is less used individually.