JAVA.IO.BUFFEREDINPUTSTREAM.SKIP METHOD

Description

The **java.io.BufferedInputStream.skip*long*** method skips n bytes of data from the buffered input stream. The number of bytes skipped id returned as **long**. For negative n, no bytes are skipped.

The skip method of **BufferedInputStream** creates a byte array which is read into until n bytes are read or the end of the stream is reached.

Declaration

Following is the declaration for **java.io.BufferedInputStream.skip*longn*** method

```
public long skip(long n)
```

Parameters

- **n** -- number of bytes to be skipped.

Return Value

Returns actual number of bytes to be skipped.

Exception

- **IOException** -- if the stream not supporting seek, or if other I/O error occurs.

Example

The following example shows the usage of java.io.BufferedInputStream.skip*longn* method.

```
package com.tutorialspoint;

import java.io.BufferedInputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

public class BufferedInputStreamDemo {
   public static void main(String[] args) throws Exception {

      InputStream is =null;
      BufferedInputStream bis = null;

      try {
         // open input stream test.txt for reading purpose.
         is = new FileInputStream("C:/test.txt");

         // input stream is converted to buffered input stream
         bis = new BufferedInputStream(is);

         // read until a single byte is available
         while(bis.available()>0)
         {
            // skip single byte from the stream
            bis.skip(1);

            // read next available byte and convert to char
            char c = (char)bis.read();

            // print character
            System.out.print(" " + c);
```

```
            }
         } catch (IOException e) {
            e.printStackTrace();
         }finally{
            // releases resources from the streams
            if(is!=null)
               is.close();
            if(bis!=null)
               bis.close();
         }
      }
   }
```

Assuming we have a text file **c:/test.txt**, which has the following content. This file will be used as an input for our example program:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Let us compile and run the above program, this will produce the following result:

```
B D F H J L N P R T V X Z
```