

# JAVA.IO.INPUTSTREAM.READ METHOD

[http://www.tutorialspoint.com/java/io/inputstream\\_read\\_byte\\_len.htm](http://www.tutorialspoint.com/java/io/inputstream_read_byte_len.htm)

Copyright © tutorialspoint.com

## Description

The **java.io.InputStream.readbyte[]b, intoff, intlen** method reads upto len bytes of data from the input stream into an array of bytes. If the parameter len is zero, then no bytes are read and 0 is returned; else there is an attempt to read atleast one byte. If the stream is at the end of the file, the value returned is -1.

## Declaration

Following is the declaration for **java.io.InputStream.readbyte[]b, intoff, intlen** method:

```
public int read(byte[] b, int off, int len)
```

## Parameters

- **b** -- The destination byte array.
- **off** -- The start offset in array b at which the data is written.
- **len** -- The number of bytes to read.

## Return Value

The method returns the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached.

## Exception

- **IOException** -- if an I/O error occurs.
- **NullPointerException** -- if b is null.
- **IndexOutOfBoundsException** -- if off is negative, len is negative, or len is greater than b.length - off.

## Example

The following example shows the usage of **java.io.InputStream.readbyte[]b, intoff, intlen** method.

```
package com.tutorialspoint;

import java.io.FileInputStream;
import java.io.InputStream;

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

        InputStream is = null;
        byte[] buffer=new byte[5];
        char c;

        try{
            // new input stream created
            is = new FileInputStream("C://test.txt");

            System.out.println("Characters printed:");

            // read stream data into buffer
            is.read(buffer, 2, 3);

            // for each byte in the buffer
```

```

for(byte b:buffer)
{
    // convert byte to character
    if(b==0)

        // if b is empty
        c='-';
    else

        // if b is read
        c=(char)b;

    // prints character
    System.out.print(c);
}
}catch(Exception e){

    // if any I/O error occurs
    e.printStackTrace();
}finally{

    // releases system resources associated with this stream
    if(is!=null)
        is.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:

```
ABCDE
```

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

```
Characters printed:
```

```
--ABC
```

```
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```