How to read file in Java - BufferedInputStream

BY CHAITANYA SINGH | FILED UNDER: JAVA I/O

In this example we will see how to read a file in Java using `FileInputStream` and `BufferedInputStream`. Here are the detailed steps that we have taken in the below code:
1) Created a File instance by providing the full path of the file(which we will read) during File Object creation.
2) Passed the file instance to the FileInputStream which opens a connection to the actual file, the file named by the File object file in the file system.
3) Passed the `FileInputStream` instance to BufferedInputStream which creates a `BufferedInputStream` and saves its argument, the input stream in, for later use. An internal buffer array is created and stored in buf using which the read operation gives good performance as the content is readily available in the buffer.
4) Used while loop to read the file. Method available() is used for checking the end of the file as it returns 0 when the pointer reaches to the end of the file. Read the file content using read() method of `FileInputStream`.

```
package beginnersbook.com;
import java.io.*;
public class ReadFileDemo {
   public static void main(String[] args) {
      //Specify the path of the file here
      File file = new File("C://myfile.txt");
      BufferedInputStream bis = null;
      FileInputStream  fis= null;

      try
      {
          //FileInputStream to read the file
          fis = new FileInputStream(file);

          /*Passed the FileInputStream to BufferedInputStream
           *For Fast read using the buffer array.*/
          bis = new BufferedInputStream(fis);

          /*available() method of BufferedInputStream
           * returns 0 when there are no more bytes
           * present in the file to be read*/
          while( bis.available() > 0 ){
              System.out.print((char)bis.read());
          }

       }catch(FileNotFoundException fnfe)
        {
            System.out.println("The specified file not found" + fnfe);
        }
        catch(IOException ioe)
```

```
        {
            System.out.println("I/O Exception: " + ioe);
        }
        finally
        {
            try{
               if(bis != null && fis!=null)
               {
                   fis.close();
                  bis.close();
               }
             }catch(IOException ioe)
              {
                  System.out.println("Error in InputStream close(): " + ioe);
              }
        }
   }
}
```