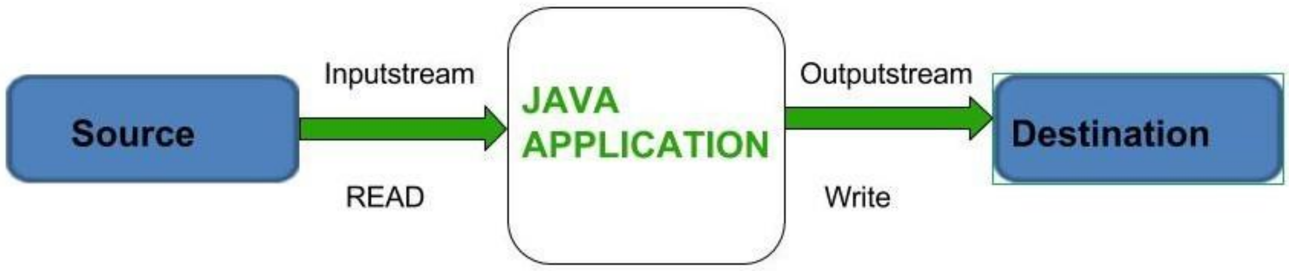
Input and output stream in java-

Java I/O (Input and Output) is used to process the input and produce the output.

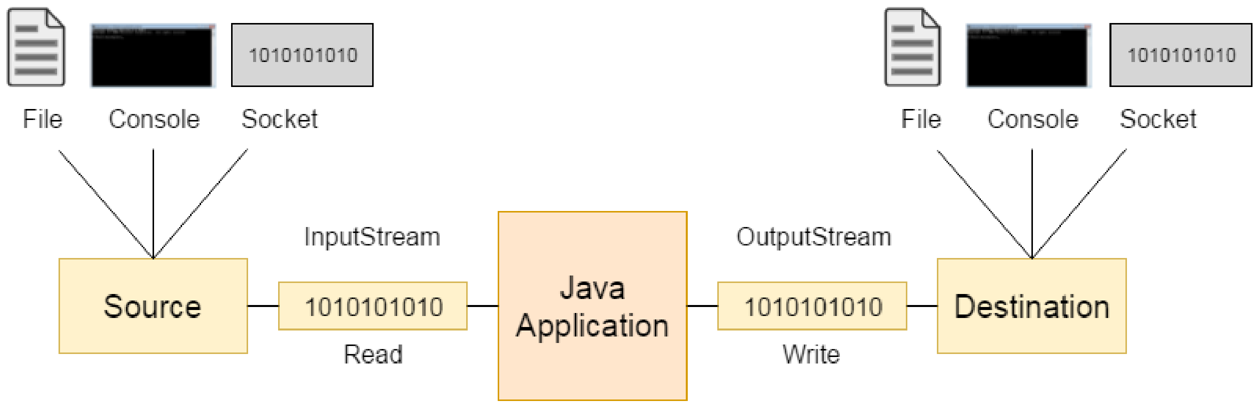


- Java uses the concept of a **stream** to make I/O operation fast. The java.io package contains all the classes required for input and output operations.
- We can perform file handling in Java by Java I/O API.

Stream

In Java, streams are the sequence of data that are read from the source and written to the destination.

An **input stream** is used to read data from the source it may be a file, an array, peripheral device or socket. And, an **output stream** is used to write data to the destination ; it may be a file, an array, peripheral device or socket.



In Java, 3 streams are created for us automatically. All these streams are attached with the console.

1) **System.out**: standard output stream [print(),println(),printf()

2) **System.in**: standard input stream

3) **System. Err**: standard error stream

Types of Streams

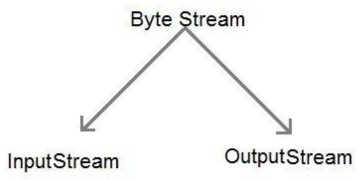
Depending upon the data a stream holds, it can be classified into:

- Byte Stream
  Binary streams have byte data that may represent a graphic or executable code, such as a Java .class file.
- Character Stream
  Text streams have character data such as an HTML file or a Java source file.

Note :A stream carries data from a source to a destination in FIFO mode.

Byte Stream:-

- Byte stream is used to read and write a single byte (8 bits) of data.
- All byte stream classes are derived from base abstract classes called InputStream and OutputStream.



There are many byte stream classes. To demonstrate how byte streams work, we'll focus on the file I/O byte streams, FileInputStream and FileOutputStream. Other kinds of byte streams are used in much the same way; they differ mainly in the way they are constructed.

Some important Byte stream classes.

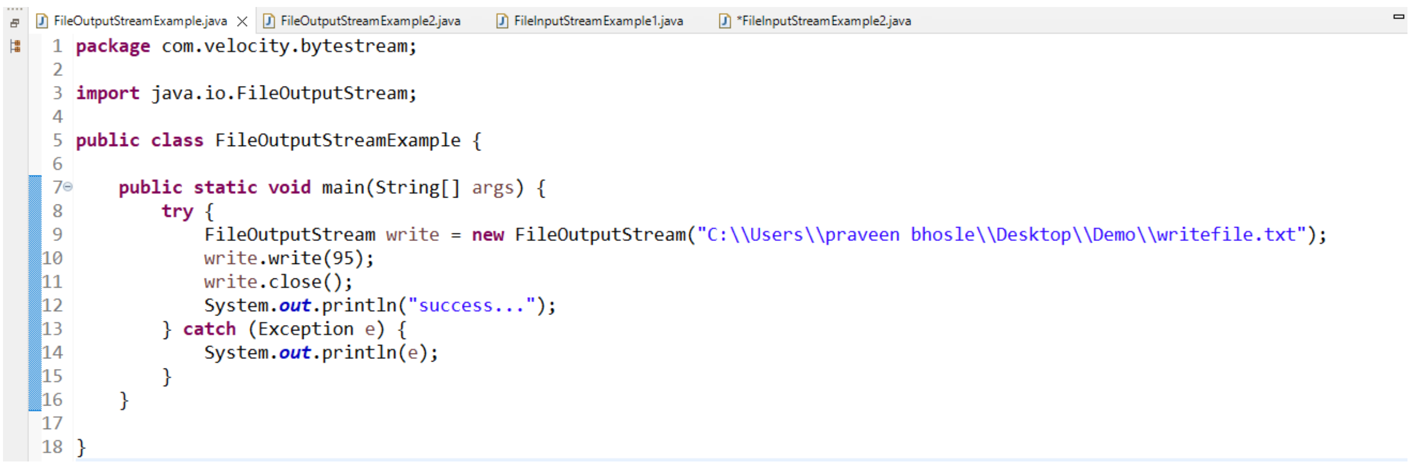
Stream class	Description
FileInputStream	Input stream that reads from a file
FileOutputStream	Output stream that write to a file.
BufferedInputStream	Used for Buffered Input Stream.
BufferedOutputStream	Used for Buffered Output Stream.
DataInputStream	Contains method for reading java standard datatype
DataOutputStream	An output stream that contain method for writing java standard data type
PrintStream	Output Stream that contain print() and println() method

These classes define several key methods. Two most important are

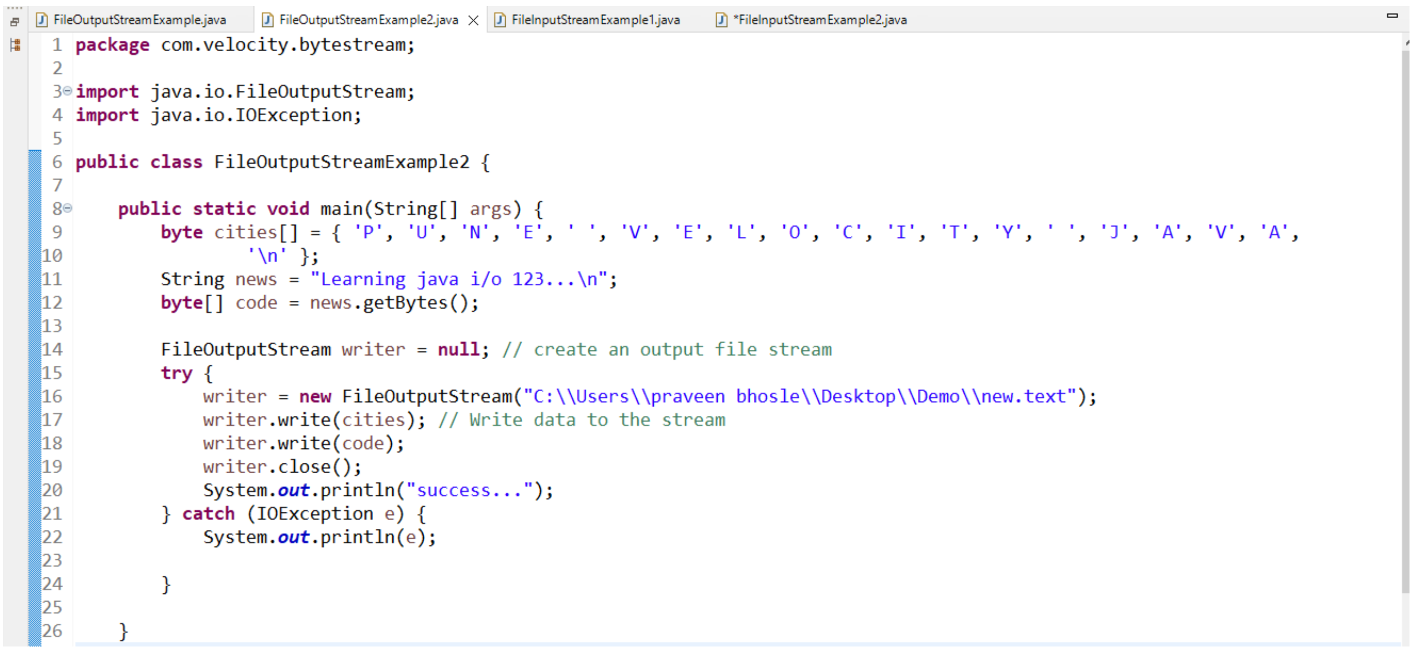
read (): reads byte of data.

write (): Writes byte of data.

Example 1:

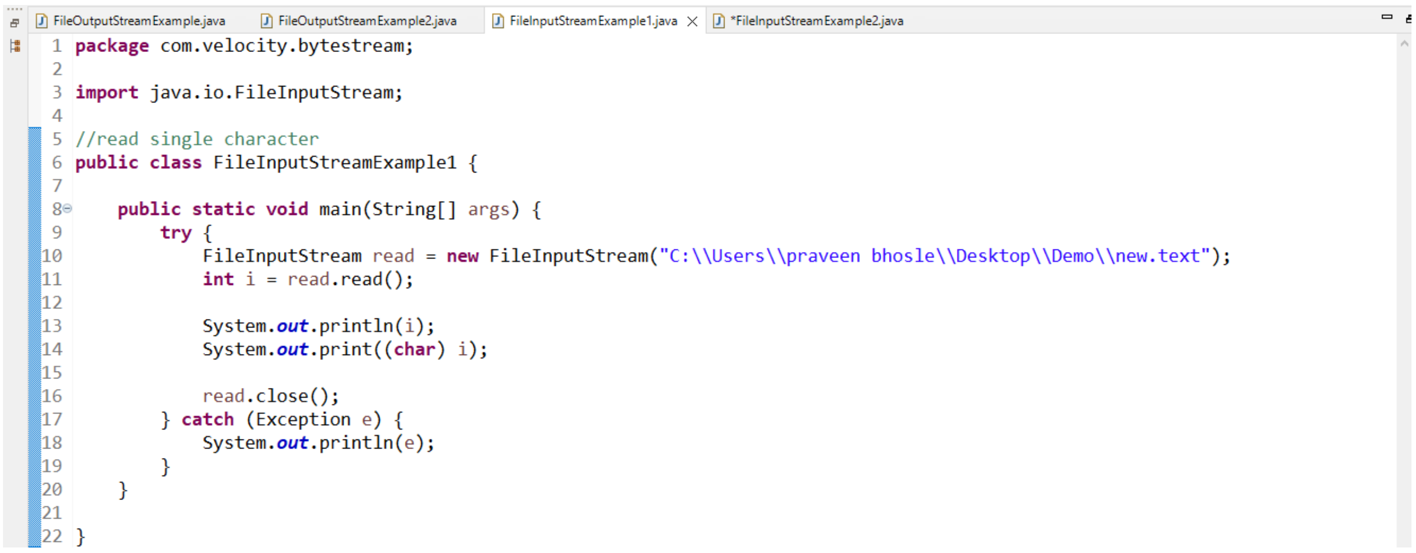

FileOutputStreamExample.java
FileOutputStreamExample2.java
FileInputStreamExample1.java
*FileInputStreamExample2.java

```java
package com.velocity.bytestream;

import java.io.FileOutputStream;

public class FileOutputStreamExample {

    public static void main(String[] args) {
        try {
            FileOutputStream write = new FileOutputStream("C:\\Users\\praveen bhosle\\Desktop\\Demo\\writefile.txt");
            write.write(95);
            write.close();
            System.out.println("success...");
        } catch (Exception e) {
            System.out.println(e);
        }
    }

}
```


FileOutputStreamExample.java
FileOutputStreamExample2.java
FileInputStreamExample1.java
*FileInputStreamExample2.java

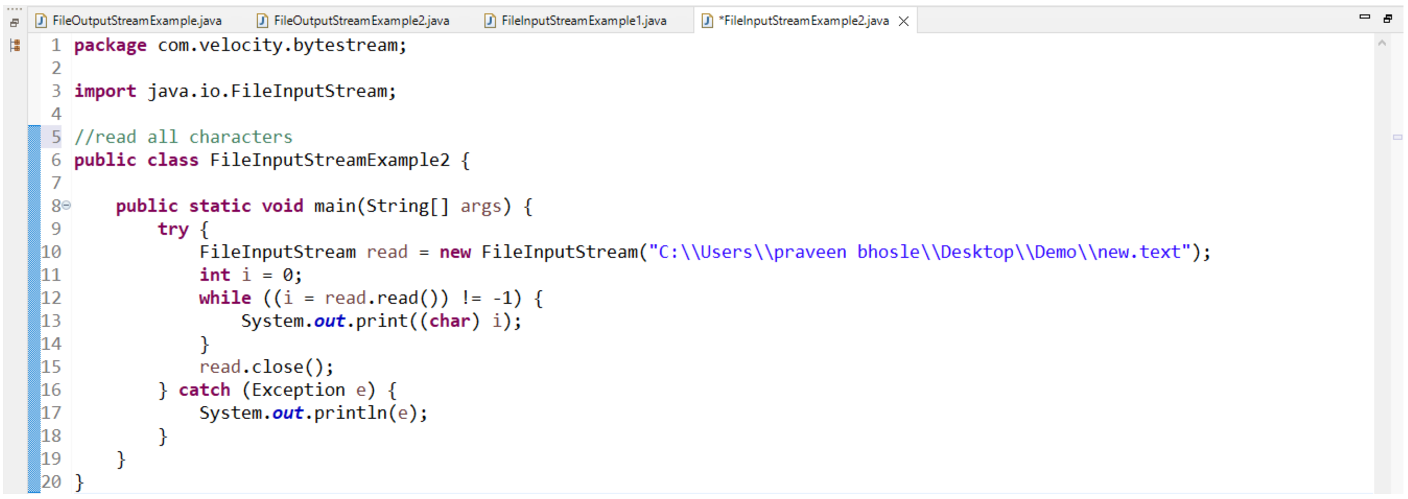
```java
package com.velocity.bytestream;

import java.io.FileOutputStream;
import java.io.IOException;

public class FileOutputStreamExample2 {

    public static void main(String[] args) {
        byte cities[] = { 'P', 'U', 'N', 'E', ' ', 'V', 'E', 'L', 'O', 'C', 'I', 'T', 'Y', ' ', 'J', 'A', 'V', 'A',
                '\n' };
        String news = "Learning java i/o 123...\n";
        byte[] code = news.getBytes();

        FileOutputStream writer = null; // create an output file stream
        try {
            writer = new FileOutputStream("C:\\Users\\praveen bhosle\\Desktop\\Demo\\new.text");
            writer.write(cities); // Write data to the stream
            writer.write(code);
            writer.close();
            System.out.println("success...");
        } catch (IOException e) {
            System.out.println(e);

        }

    }
```

Example 2:



FileOutputStreamExample.java
FileOutputStreamExample2.java
FileInputStreamExample1.java
*FileInputStreamExample2.java

```java
package com.velocity.bytestream;

import java.io.FileInputStream;

//read single character
public class FileInputStreamExample1 {

    public static void main(String[] args) {
        try {
            FileInputStream read = new FileInputStream("C:\\Users\\praveen bhosle\\Desktop\\Demo\\new.text");
            int i = read.read();

            System.out.println(i);
            System.out.print((char) i);

            read.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }

}
```

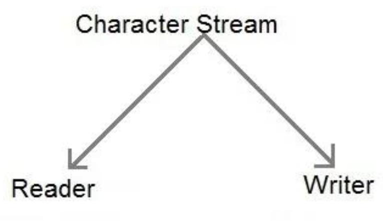


FileOutputStreamExample.java
FileOutputStreamExample2.java
FileInputStreamExample1.java
*FileInputStreamExample2.java

```java
package com.velocity.bytestream;

import java.io.FileInputStream;

//read all characters
public class FileInputStreamExample2 {

    public static void main(String[] args) {
        try {
            FileInputStream read = new FileInputStream("C:\\Users\\praveen bhosle\\Desktop\\Demo\\new.text");
            int i = 0;
            while ((i = read.read()) != -1) {
                System.out.print((char) i);
            }
            read.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Character Stream:

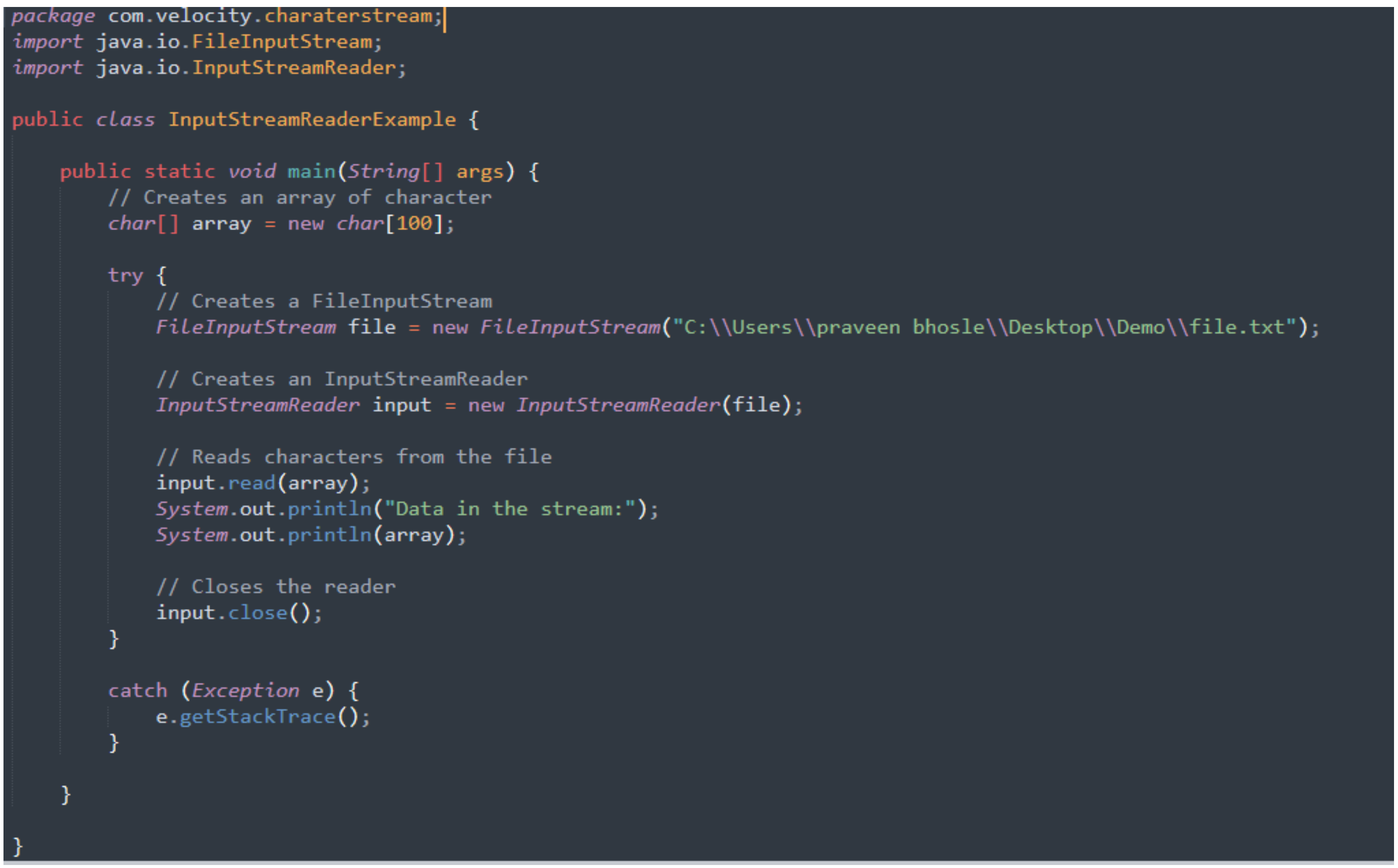
- Character stream is used to read and write a single character of data.
- All the character stream classes are derived from base abstract classes Reader and Writer.



Some important Character stream classes

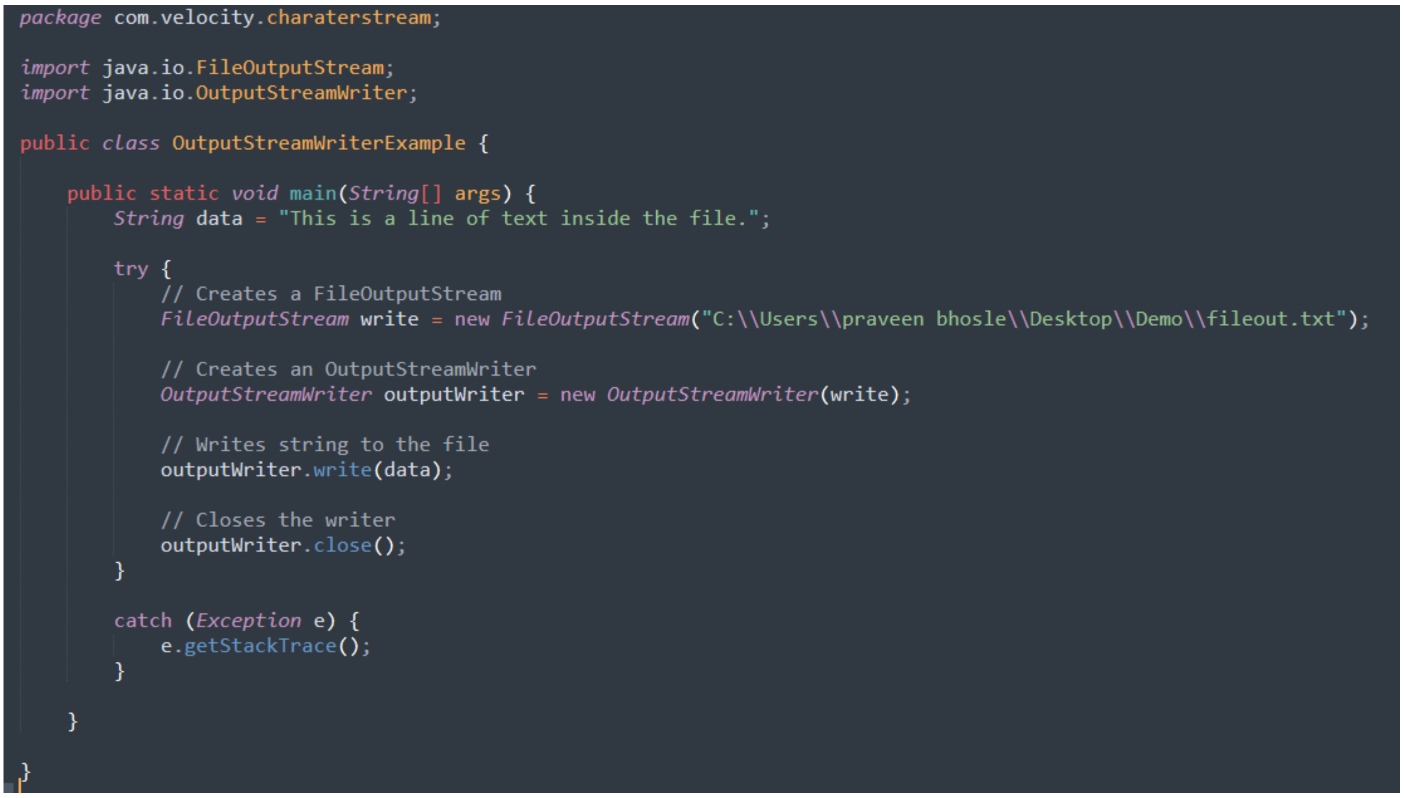
Stream class	Description
BufferedReader	Handles buffered input stream.
BufferedWriter	Handles buffered output stream.
FileReader	Input stream that reads from file.
FileWriter	Output stream that writes to file.
InputStreamReader	Input stream that translate byte to character
OutputStreamReader	Output stream that translate character to byte.
PrintWriter	Output Stream that contain print() and println() method.

#Program 1 : InputStreamReader class



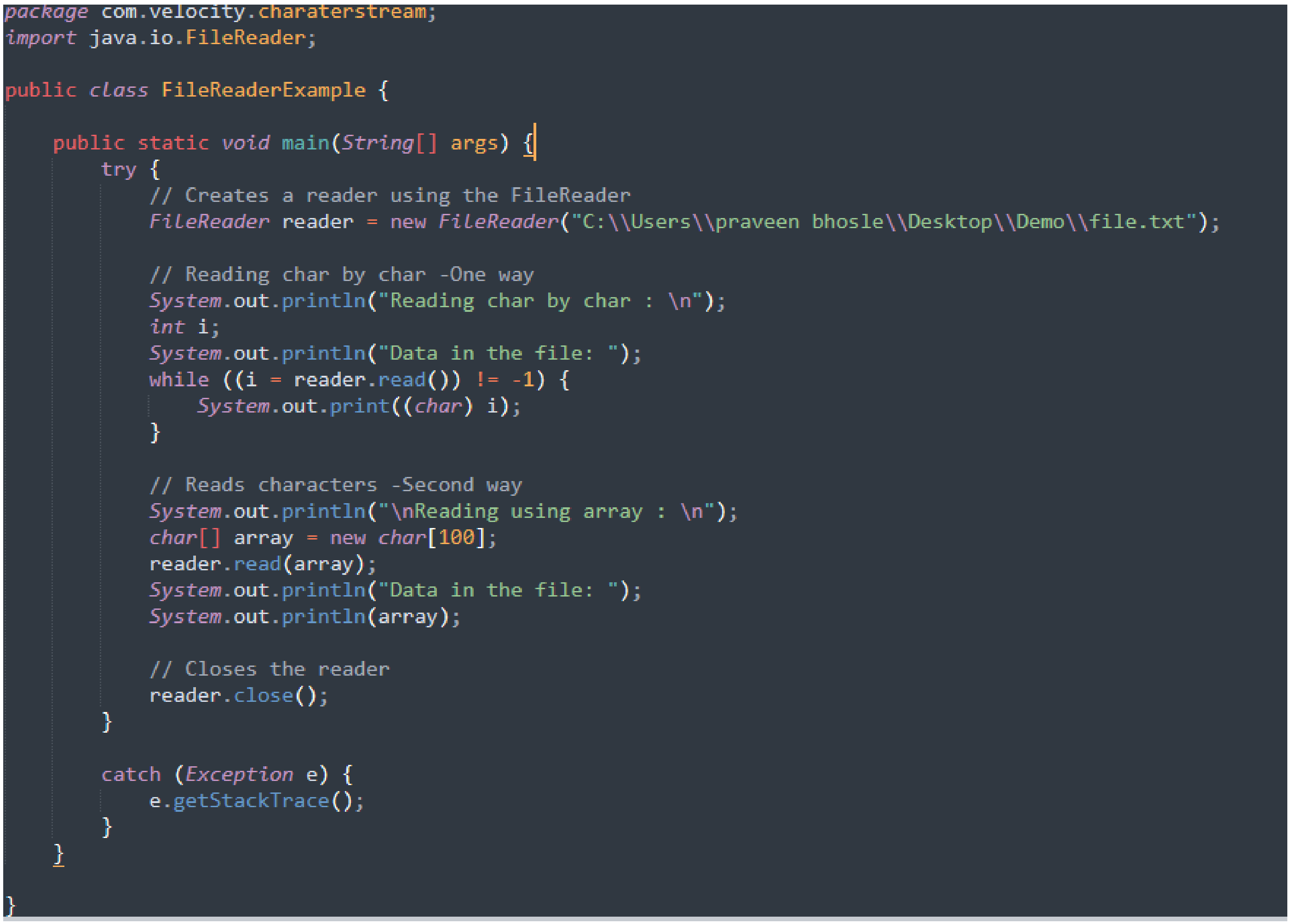
```java
package com.velocity.charaterstream;
import java.io.FileInputStream;
import java.io.InputStreamReader;

public class InputStreamReaderExample {

    public static void main(String[] args) {
        // Creates an array of character
        char[] array = new char[100];

        try {
            // Creates a FileInputStream
            FileInputStream file = new FileInputStream("C:\\Users\\praveen bhosle\\Desktop\\Demo\\file.txt");

            // Creates an InputStreamReader
            InputStreamReader input = new InputStreamReader(file);

            // Reads characters from the file
            input.read(array);
            System.out.println("Data in the stream:");
            System.out.println(array);

            // Closes the reader
            input.close();
        }

        catch (Exception e) {
            e.getStackTrace();
        }

    }

}
```

Program 2 : OutPutStreamWriter class



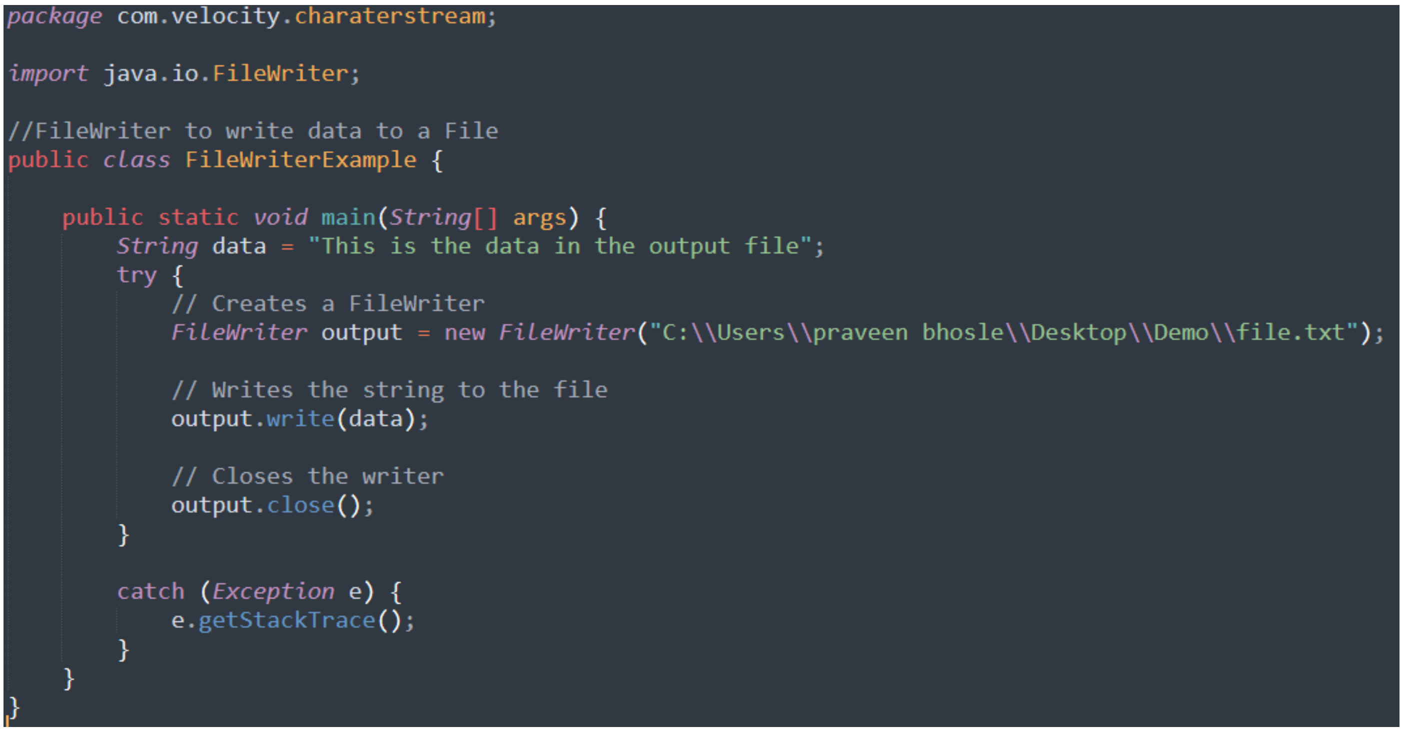
```java
package com.velocity.charaterstream;

import java.io.FileOutputStream;
import java.io.OutputStreamWriter;

public class OutputStreamWriterExample {

    public static void main(String[] args) {
        String data = "This is a line of text inside the file.";

        try {
            // Creates a FileOutputStream
            FileOutputStream write = new FileOutputStream("C:\\Users\\praveen bhosle\\Desktop\\Demo\\fileout.txt");

            // Creates an OutputStreamWriter
            OutputStreamWriter outputWriter = new OutputStreamWriter(write);

            // Writes string to the file
            outputWriter.write(data);

            // Closes the writer
            outputWriter.close();
        }

        catch (Exception e) {
            e.getStackTrace();
        }

    }

}
```

#Program 3 : FileReader class



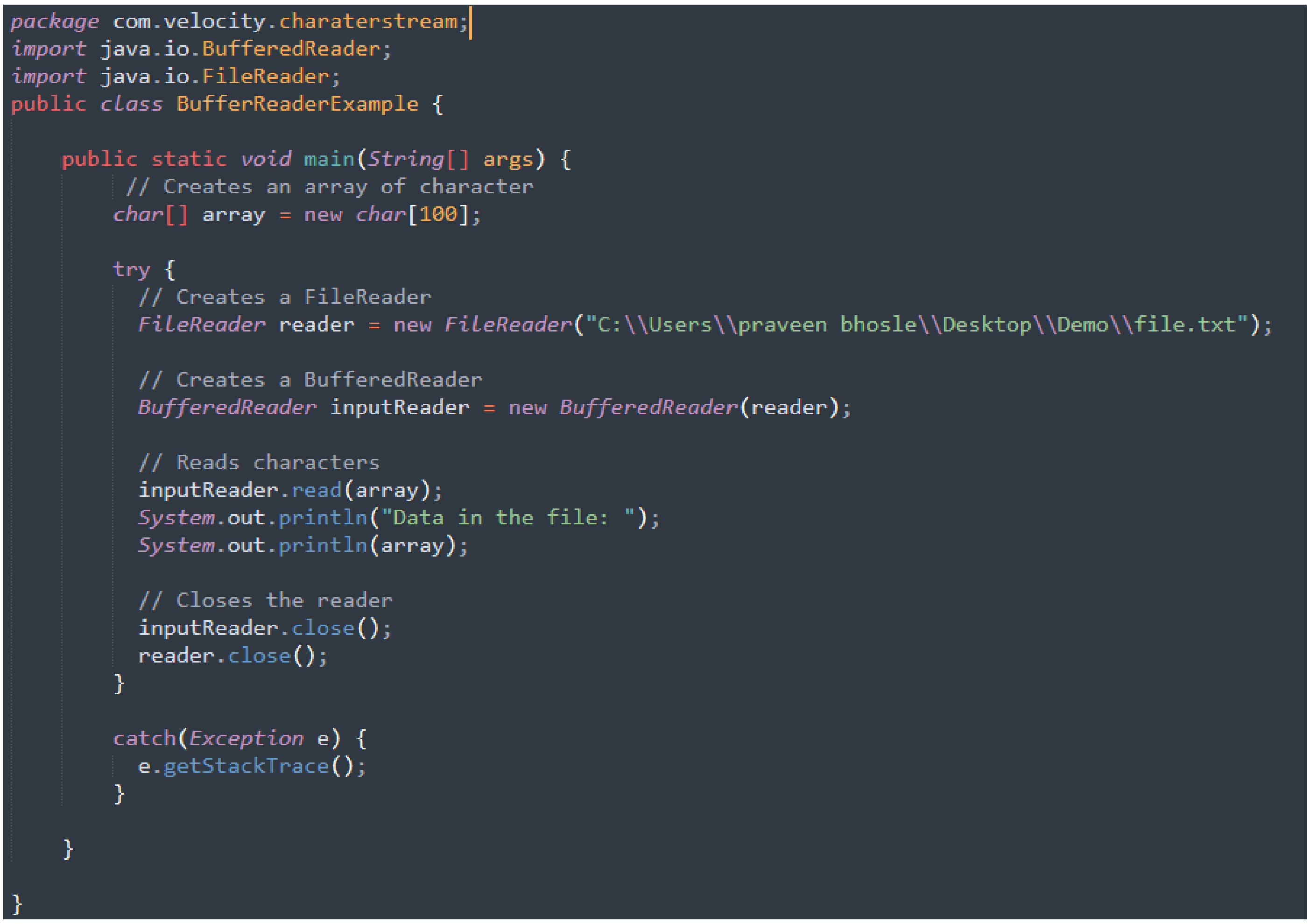
```java
package com.velocity.charaterstream;
import java.io.FileReader;

public class FileReaderExample {

    public static void main(String[] args) {
        try {
            // Creates a reader using the FileReader
            FileReader reader = new FileReader("C:\\Users\\praveen bhosle\\Desktop\\Demo\\file.txt");

            // Reading char by char -One way
            System.out.println("Reading char by char : \n");
            int i;
            System.out.println("Data in the file: ");
            while ((i = reader.read()) != -1) {
                System.out.print((char) i);
            }

            // Reads characters -Second way
            System.out.println("\nReading using array : \n");
            char[] array = new char[100];
            reader.read(array);
            System.out.println("Data in the file: ");
            System.out.println(array);

            // Closes the reader
            reader.close();
        }

        catch (Exception e) {
            e.getStackTrace();
        }
    }

}
```

#Program-4 : FileWriter class



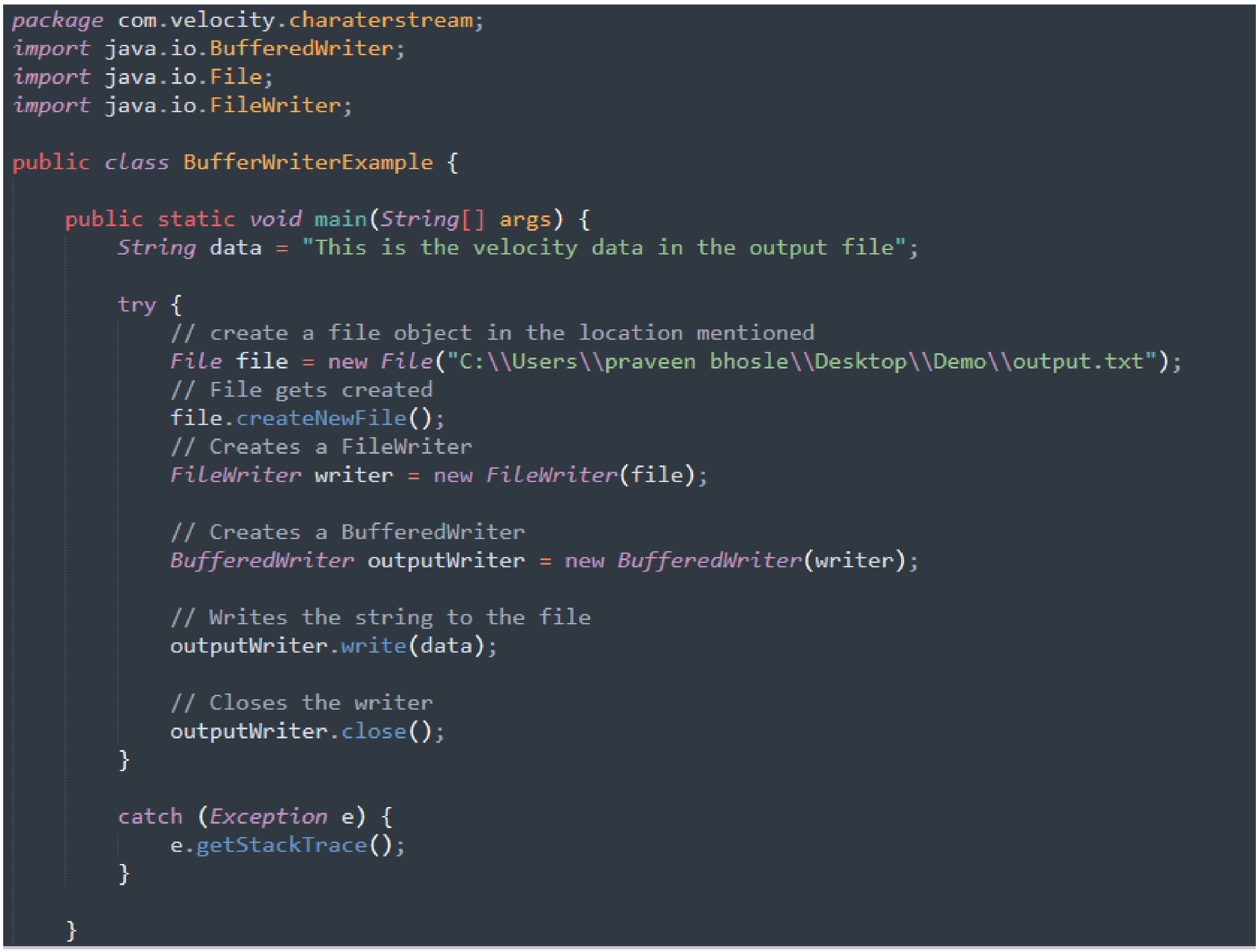
```
package com.velocity.charaterstream;

import java.io.FileWriter;

//FileWriter to write data to a File
public class FileWriterExample {

    public static void main(String[] args) {
        String data = "This is the data in the output file";
        try {
            // Creates a FileWriter
            FileWriter output = new FileWriter("C:\\Users\\praveen bhosle\\Desktop\\Demo\\file.txt");

            // Writes the string to the file
            output.write(data);

            // Closes the writer
            output.close();
        }

        catch (Exception e) {
            e.getStackTrace();
        }
    }
}
```

#Program-5: BufferReader class



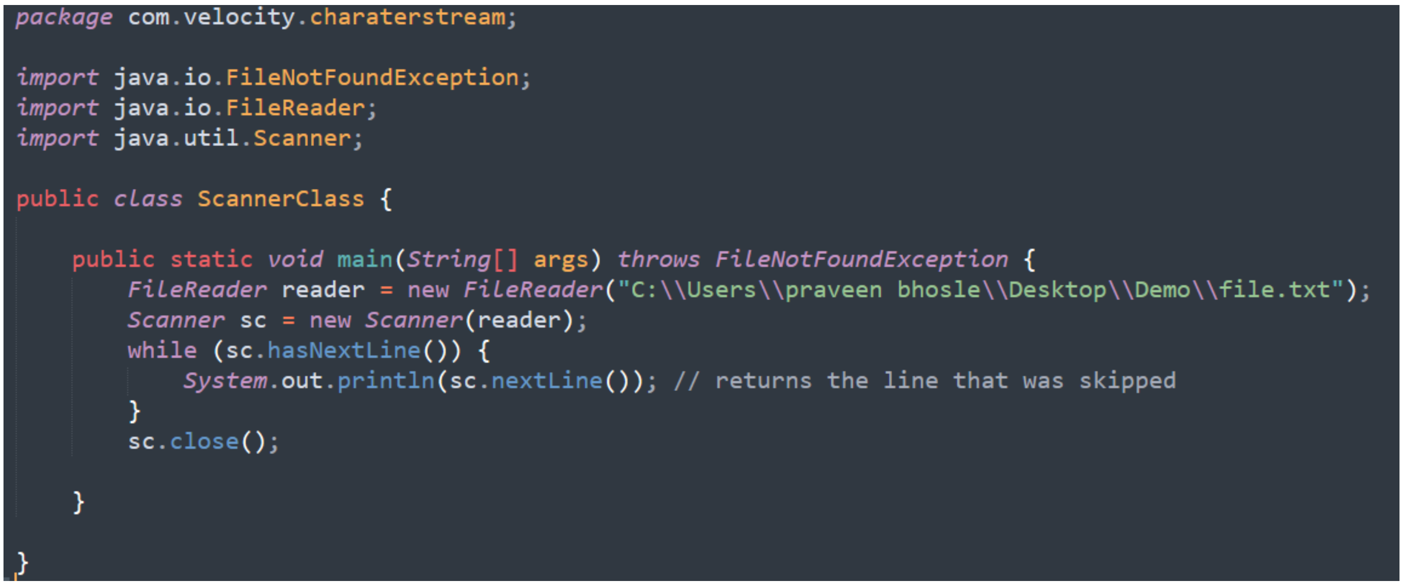
```
package com.velocity.charaterstream;
import java.io.BufferedReader;
import java.io.FileReader;
public class BufferReaderExample {

    public static void main(String[] args) {
        // Creates an array of character
        char[] array = new char[100];

        try {
          // Creates a FileReader
          FileReader reader = new FileReader("C:\\Users\\praveen bhosle\\Desktop\\Demo\\file.txt");

          // Creates a BufferedReader
          BufferedReader inputReader = new BufferedReader(reader);

          // Reads characters
          inputReader.read(array);
          System.out.println("Data in the file: ");
          System.out.println(array);

          // Closes the reader
          inputReader.close();
          reader.close();
        }

        catch(Exception e) {
          e.getStackTrace();
        }

    }

}
```

#Program6 : BufferWriter class



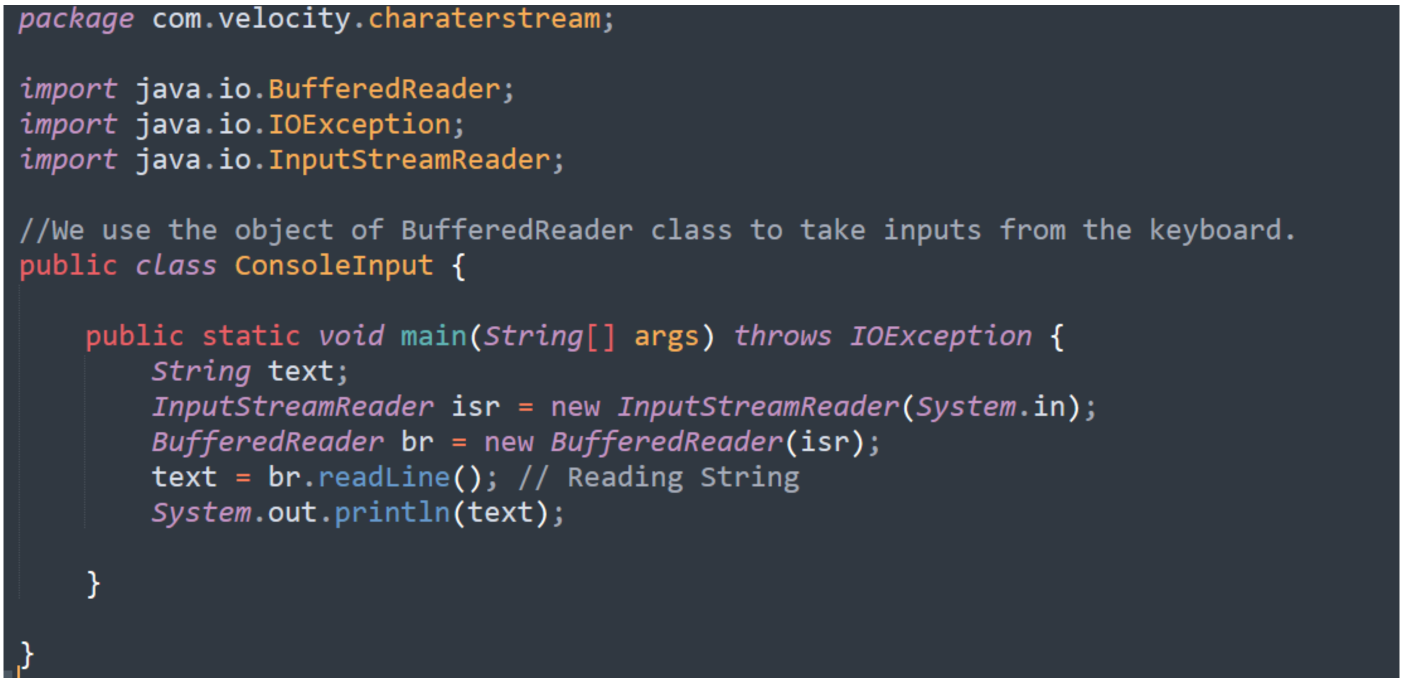
```java
package com.velocity.charaterstream;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;

public class BufferWriterExample {

    public static void main(String[] args) {
        String data = "This is the velocity data in the output file";

        try {
            // create a file object in the location mentioned
            File file = new File("C:\\Users\\praveen bhosle\\Desktop\\Demo\\output.txt");
            // File gets created
            file.createNewFile();
            // Creates a FileWriter
            FileWriter writer = new FileWriter(file);

            // Creates a BufferedWriter
            BufferedWriter outputWriter = new BufferedWriter(writer);

            // Writes the string to the file
            outputWriter.write(data);

            // Closes the writer
            outputWriter.close();
        }

        catch (Exception e) {
            e.getStackTrace();
        }

    }
```

#Program 7 : Scanner class



```java
package com.velocity.charaterstream;

import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;

public class ScannerClass {

    public static void main(String[] args) throws FileNotFoundException {
        FileReader reader = new FileReader("C:\\Users\\praveen bhosle\\Desktop\\Demo\\file.txt");
        Scanner sc = new Scanner(reader);
        while (sc.hasNextLine()) {
            System.out.println(sc.nextLine()); // returns the line that was skipped
        }
        sc.close();

    }

}
```

#Program-8 : How to take console input.



```java
package com.velocity.charaterstream;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

//We use the object of BufferedReader class to take inputs from the keyboard.
public class ConsoleInput {

    public static void main(String[] args) throws IOException {
        String text;
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(isr);
        text = br.readLine(); // Reading String
        System.out.println(text);

    }

}
```