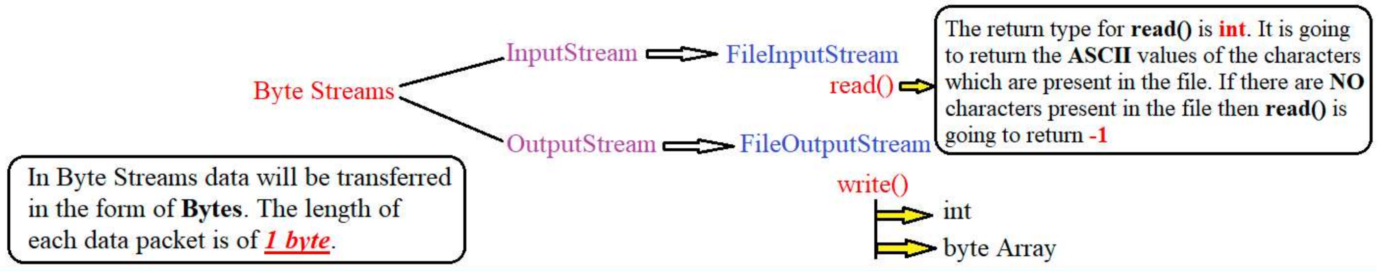
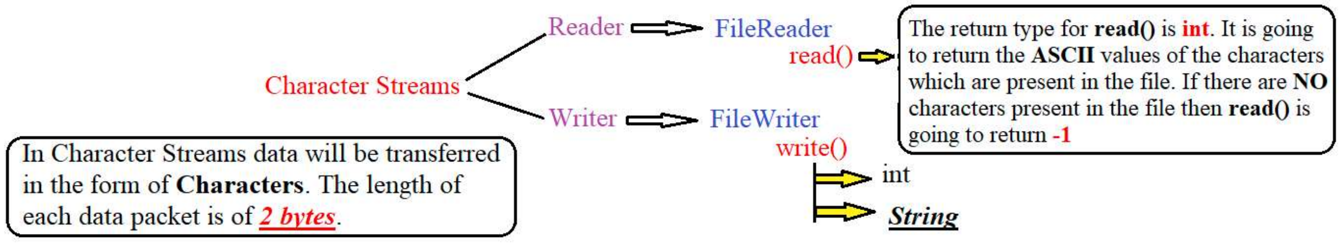


The return type for **read()** is **int**. It is going to return the **ASCII** values of the characters which are present in the file. If there are **NO** characters present in the file then **read()** is going to return **-1**

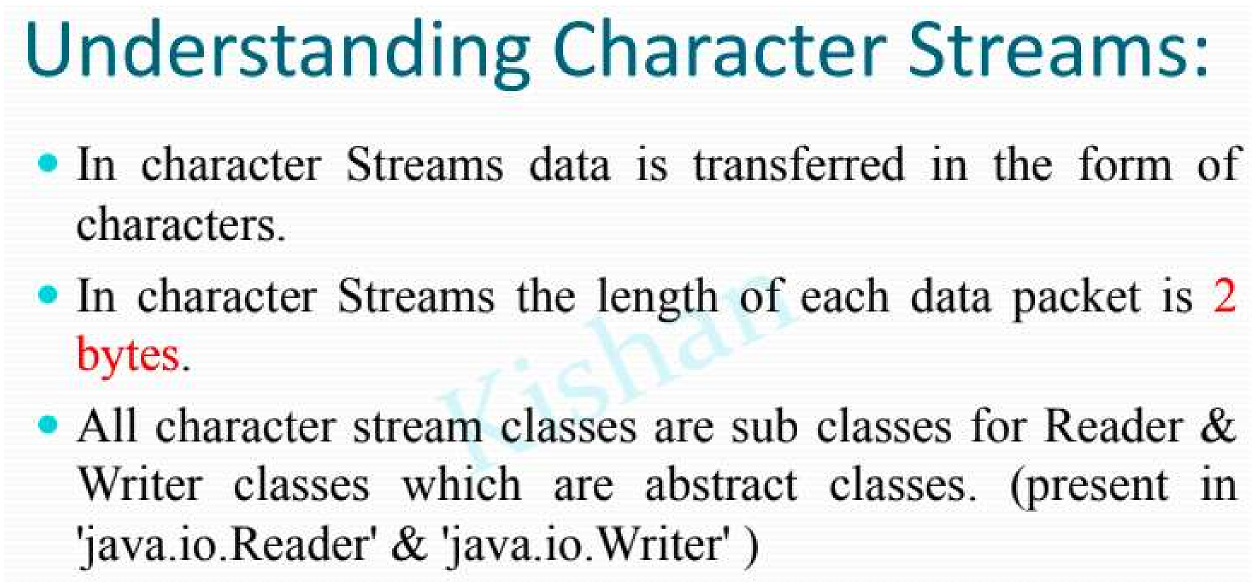
In Byte Streams data will be transferred in the form of **Bytes**. The length of each data packet is of ***1 byte***.



The return type for **read()** is **int**. It is going to return the **ASCII** values of the characters which are present in the file. If there are **NO** characters present in the file then **read()** is going to return **-1**

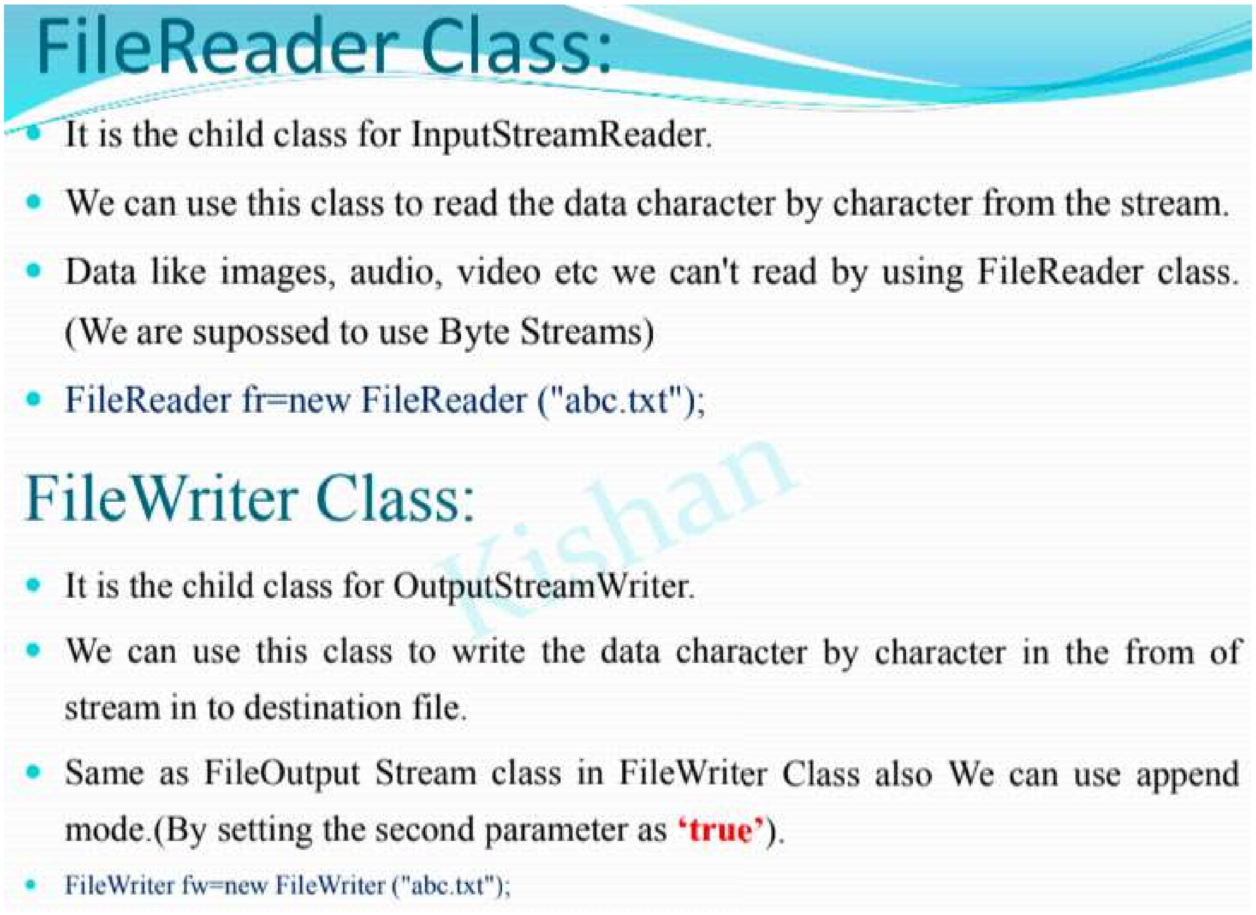
In Character Streams data will be transferred in the form of **Characters**. The length of each data packet is of ***2 bytes***.

Reader class and Writer class are abstract classes



Understanding Character Streams:

- In character Streams data is transferred in the form of characters.
- In character Streams the length of each data packet is 2 bytes.
- All character stream classes are sub classes for Reader & Writer classes which are abstract classes. (present in 'java.io.Reader' & 'java.io.Writer')



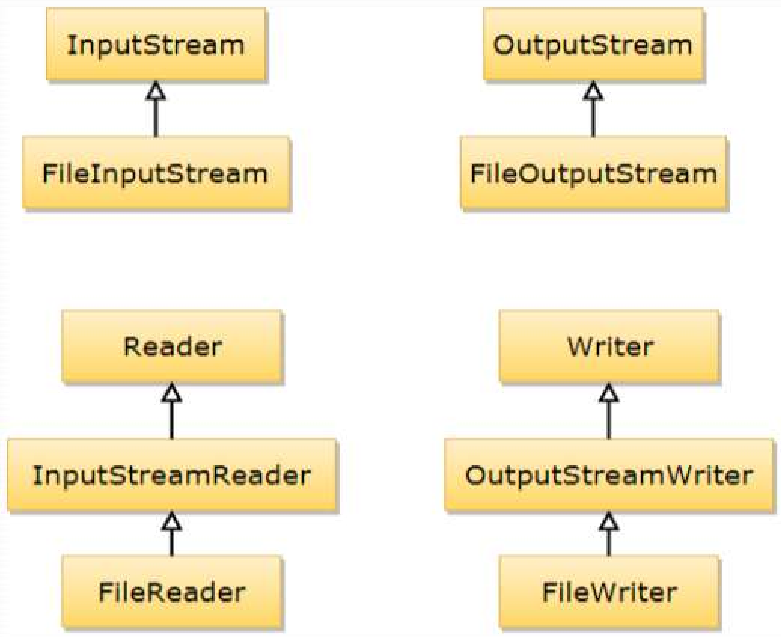
FileReader Class:

- It is the child class for InputStreamReader.
- We can use this class to read the data character by character from the stream.
- Data like images, audio, video etc we can't read by using FileReader class. (We are supossed to use Byte Streams)
- FileReader fr=new FileReader ("abc.txt");

FileWriter Class:

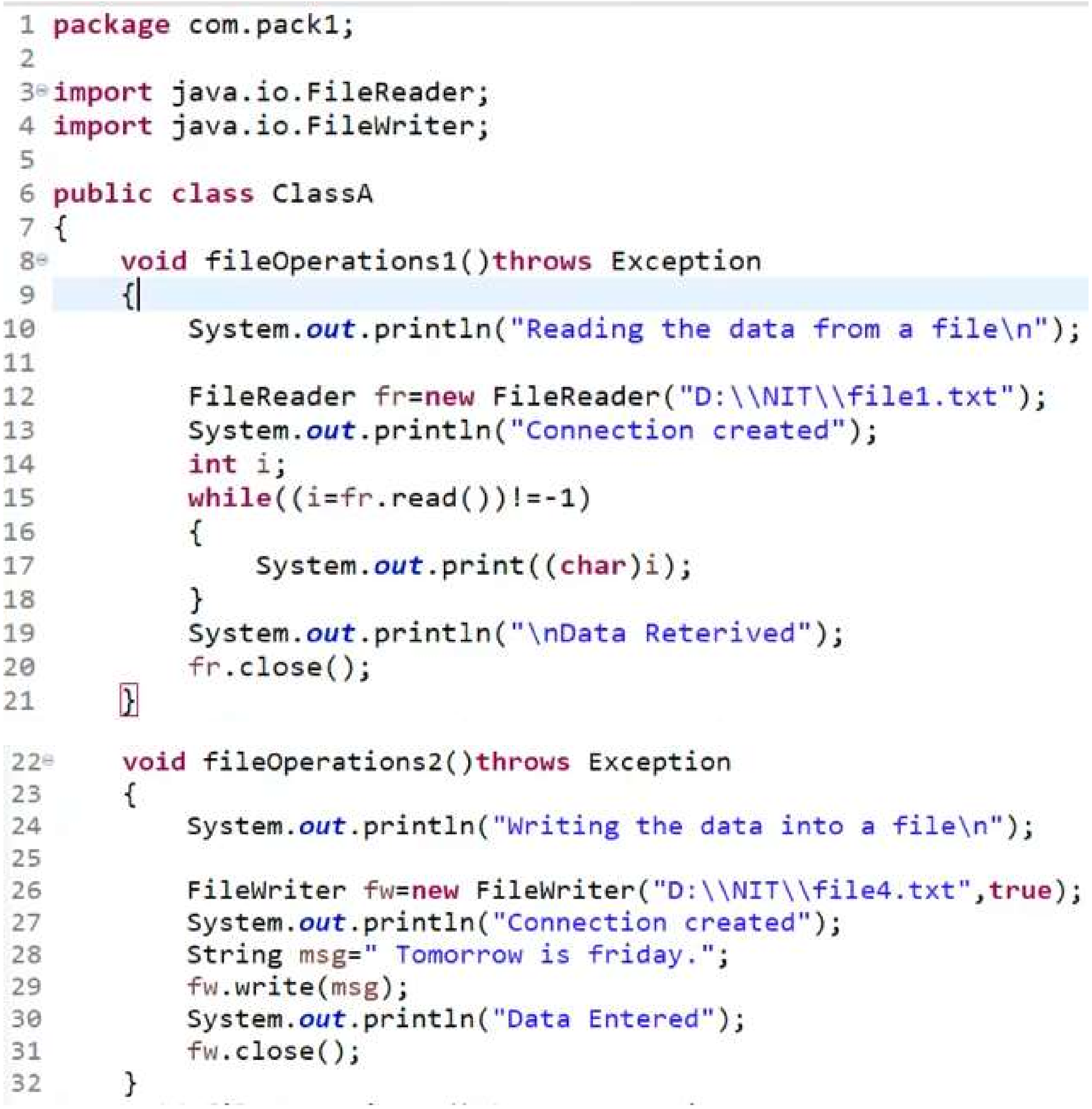
- It is the child class for OutputStreamWriter.
- We can use this class to write the data character by character in the from of stream in to destination file.
- Same as FileOutput Stream class in FileWriter Class also We can use append mode.(By setting the second parameter as **'true'**).
- FileWriter fw=new FileWriter ("abc.txt");

Byte Streams Vs Character Streams

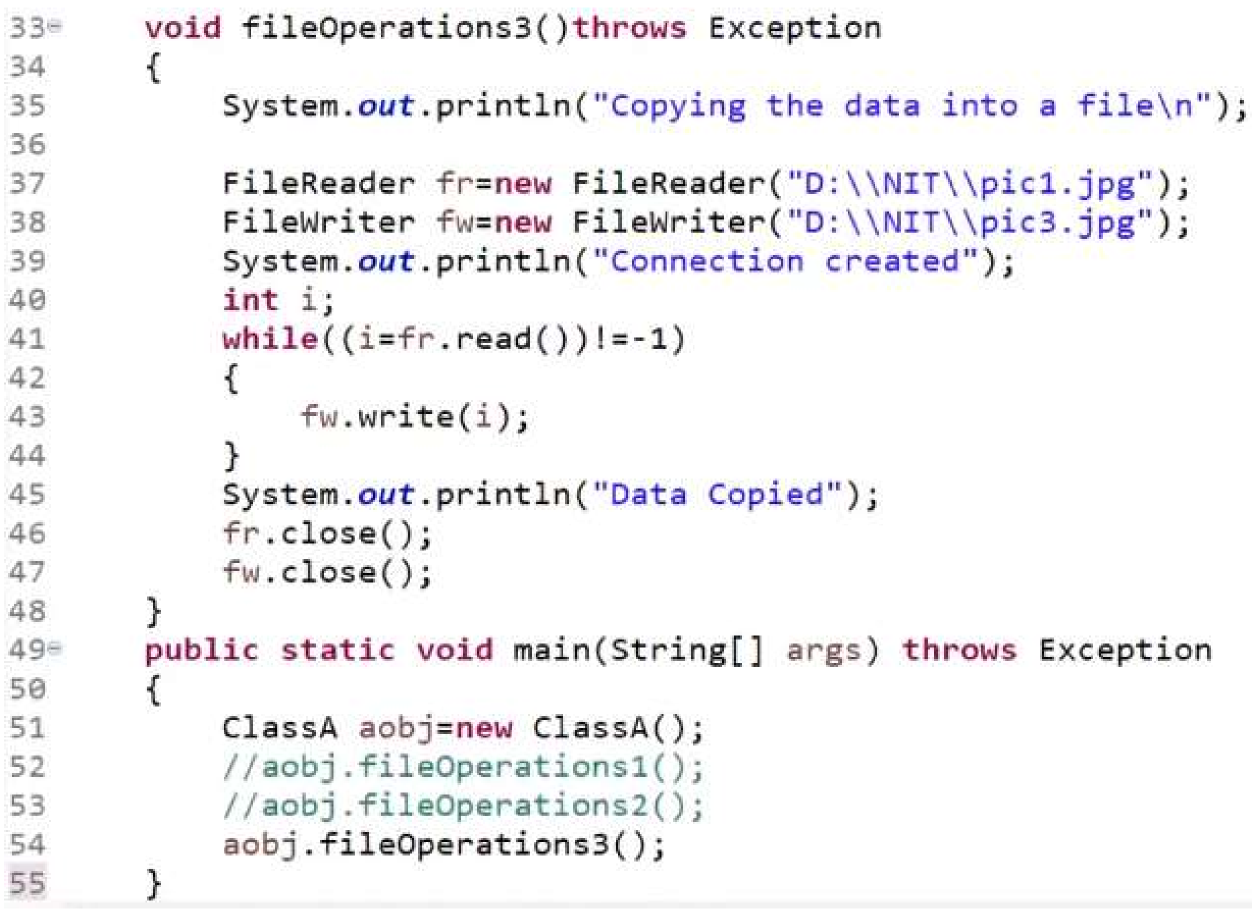


By using byte streams we can transfer both textual and multimedia data (Images, videos, audio   etc.)

By using character streams, we can transfer only textual and but not multimedia data (Images, videos, audio etc.) if we transfer the file is created with empty (no data).



```java
package com.pack1;

import java.io.FileReader;
import java.io.FileWriter;

public class ClassA
{
    void fileOperations1()throws Exception
    {
        System.out.println("Reading the data from a file\n");

        FileReader fr=new FileReader("D:\\NIT\\file1.txt");
        System.out.println("Connection created");
        int i;
        while((i=fr.read())!=-1)
        {
            System.out.print((char)i);
        }
        System.out.println("\nData Reterived");
        fr.close();
    }
    void fileOperations2()throws Exception
    {
        System.out.println("Writing the data into a file\n");

        FileWriter fw=new FileWriter("D:\\NIT\\file4.txt",true);
        System.out.println("Connection created");
        String msg=" Tomorrow is friday.";
        fw.write(msg);
        System.out.println("Data Entered");
        fw.close();
    }
```



```java
    void fileOperations3()throws Exception
    {
        System.out.println("Copying the data into a file\n");

        FileReader fr=new FileReader("D:\\NIT\\pic1.jpg");
        FileWriter fw=new FileWriter("D:\\NIT\\pic3.jpg");
        System.out.println("Connection created");
        int i;
        while((i=fr.read())!=-1)
        {
            fw.write(i);
        }
        System.out.println("Data Copied");
        fr.close();
        fw.close();
    }
    public static void main(String[] args) throws Exception
    {
        ClassA aobj=new ClassA();
        //aobj.fileOperations1();
        //aobj.fileOperations2();
        aobj.fileOperations3();
    }
```

Understanding Buffered Streams:

- A Buffer is a portion in the memory that is used to store a stream of data.
- In I/O operations each read or write request is handled directly by the underlying OS.
- This can make a program much less efficient, since each such request often triggers disk access, network activity, or some other operation that is relatively expensive.
- To reduce this kind of overhead, the Java platform implemented buffered I/O streams.
- Buffered input streams read data from a memory area known as a buffer.
- Buffered output streams write data to a buffer.
- Buffered streams are same like Byte & Character Streams but with more efficiency.

- There are four buffered stream classes used to wrap unbuffered streams
- BufferedInputStream and BufferedOutputStream create buffered byte streams
- BufferedReader and BufferedWriter create buffered character streams.

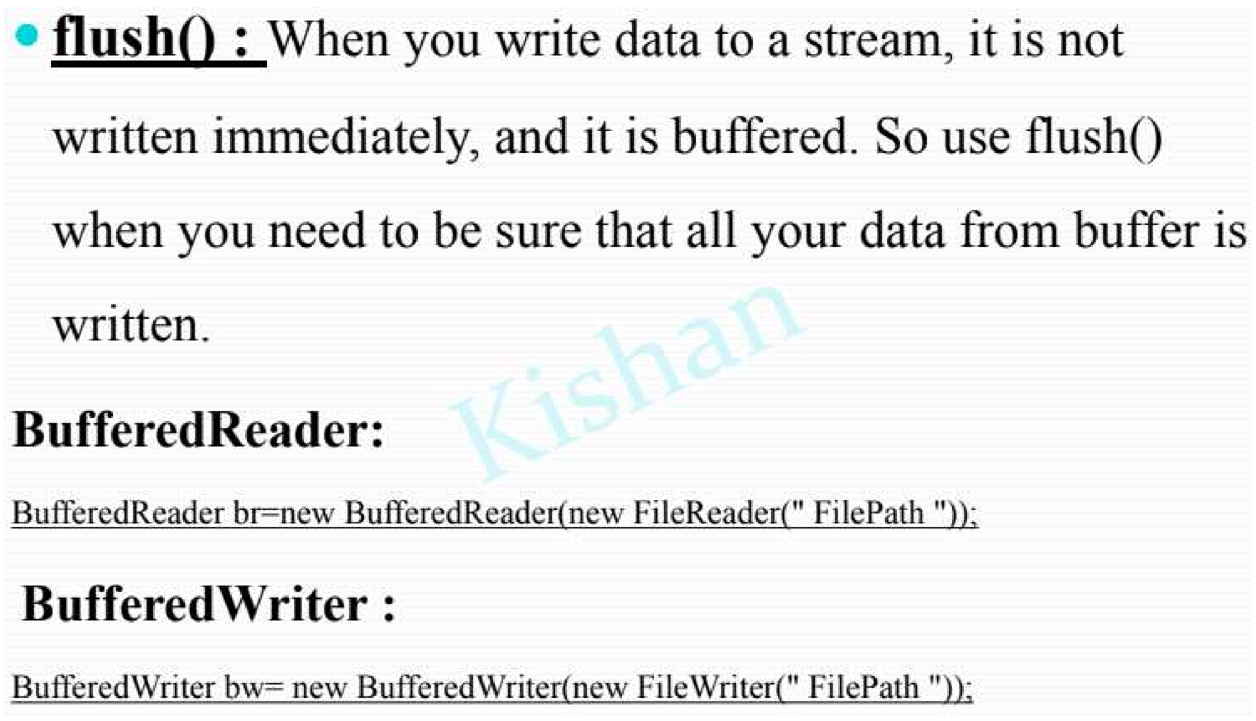
Syntax:

BufferedInputStream:

```
BufferedInputStream br=new BufferedInputStream(new FileInputStream("FilePath"));
```

BufferedOutputStream:

```
BufferedOutputStream br=new BufferedOutputStream(new FileOutputStream("FilePath"));
```



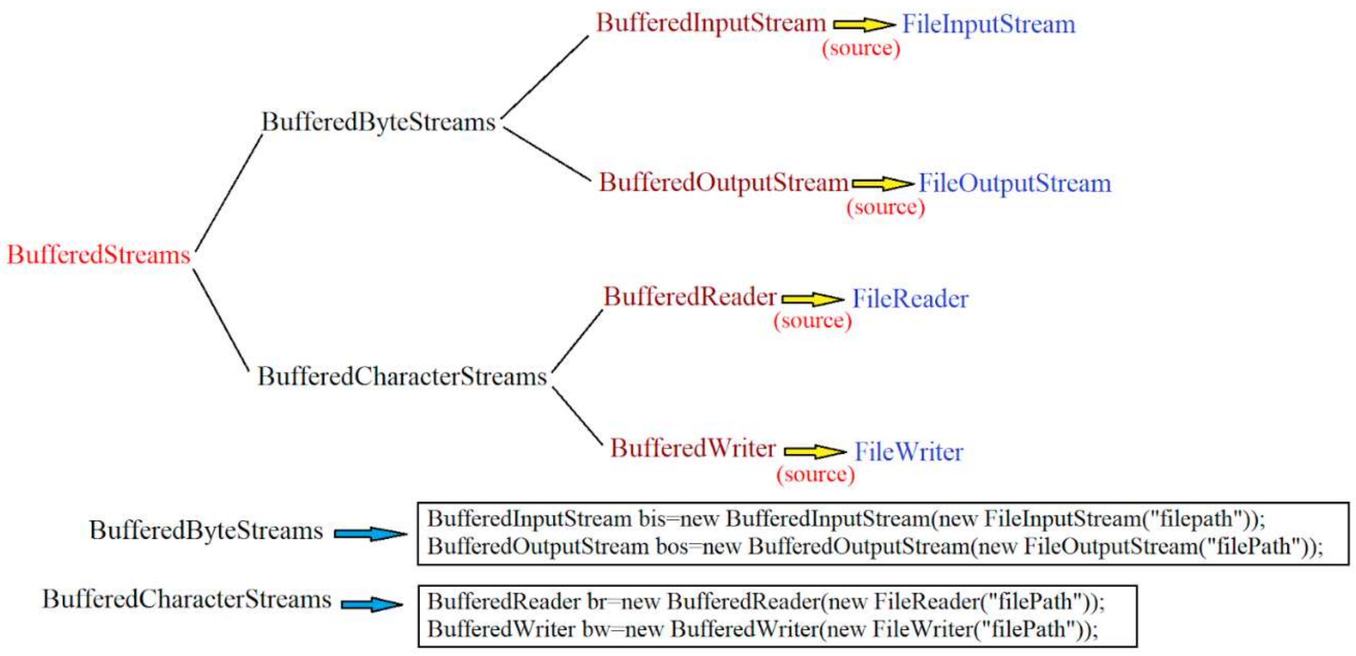
- **flush() :** When you write data to a stream, it is not written immediately, and it is buffered. So use flush() when you need to be sure that all your data from buffer is written.

BufferedReader:

BufferedReader br=new BufferedReader(new FileReader(" FilePath "));

BufferedWriter :

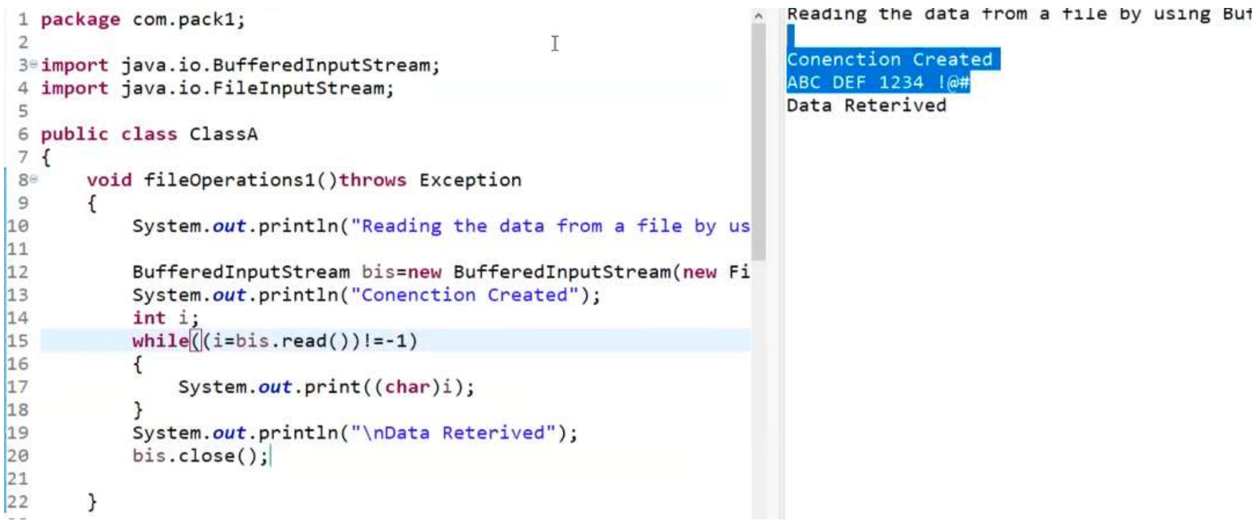
BufferedWriter bw= new BufferedWriter(new FileWriter(" FilePath "));



```
BufferedInputStream bis=new BufferedInputStream(new FileInputStream("filepath"));
BufferedOutputStream bos=new BufferedOutputStream(new FileOutputStream("filePath"));
```

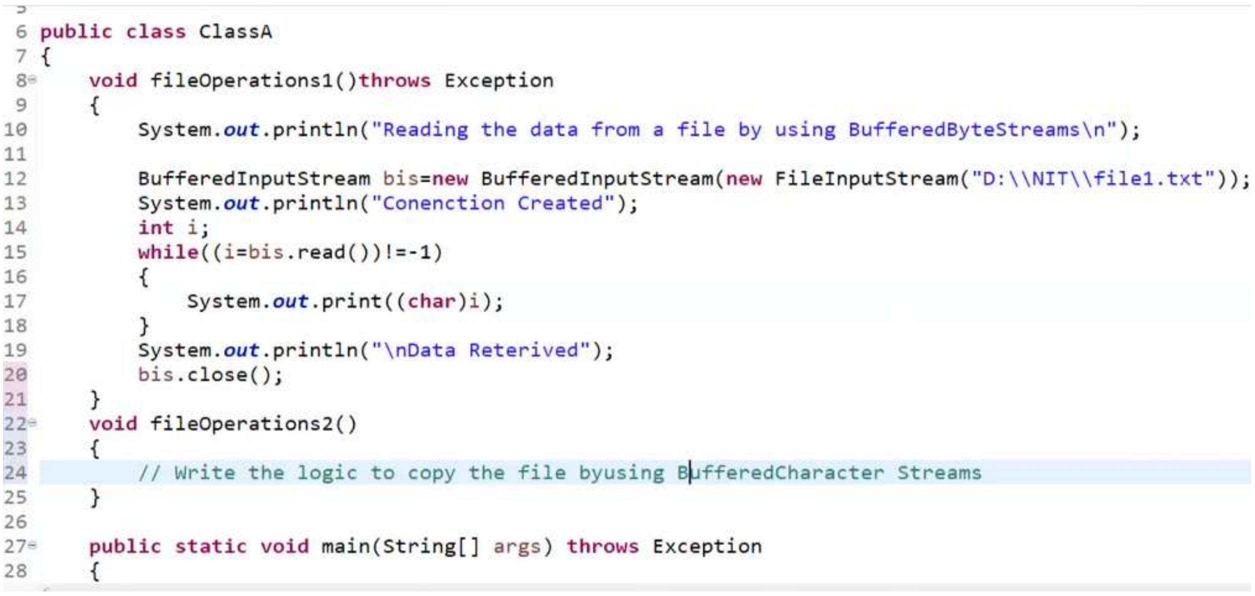
```
BufferedReader br=new BufferedReader(new FileReader("filePath"));
BufferedWriter bw=new BufferedWriter(new FileWriter("filePath"));
```

BufferInputStream and BufferOutputStream are implementation classes.

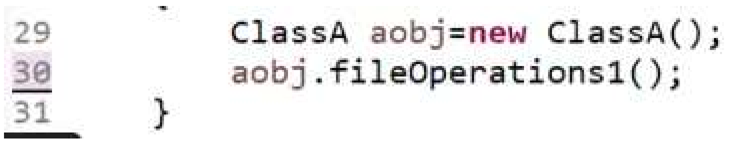


```
package com.pack1;

import java.io.BufferedInputStream;
import java.io.FileInputStream;

public class ClassA
{
    void fileOperations1()throws Exception
    {
        System.out.println("Reading the data from a file by us

        BufferedInputStream bis=new BufferedInputStream(new Fi
        System.out.println("Conenction Created");
        int i;
        while((i=bis.read())!=-1)
        {
            System.out.print((char)i);
        }
        System.out.println("\nData Reterived");
        bis.close();

    }
```

```
Reading the data from a file by using Buf

Conenction Created
ABC DEF 1234 !@#
Data Reterived
```

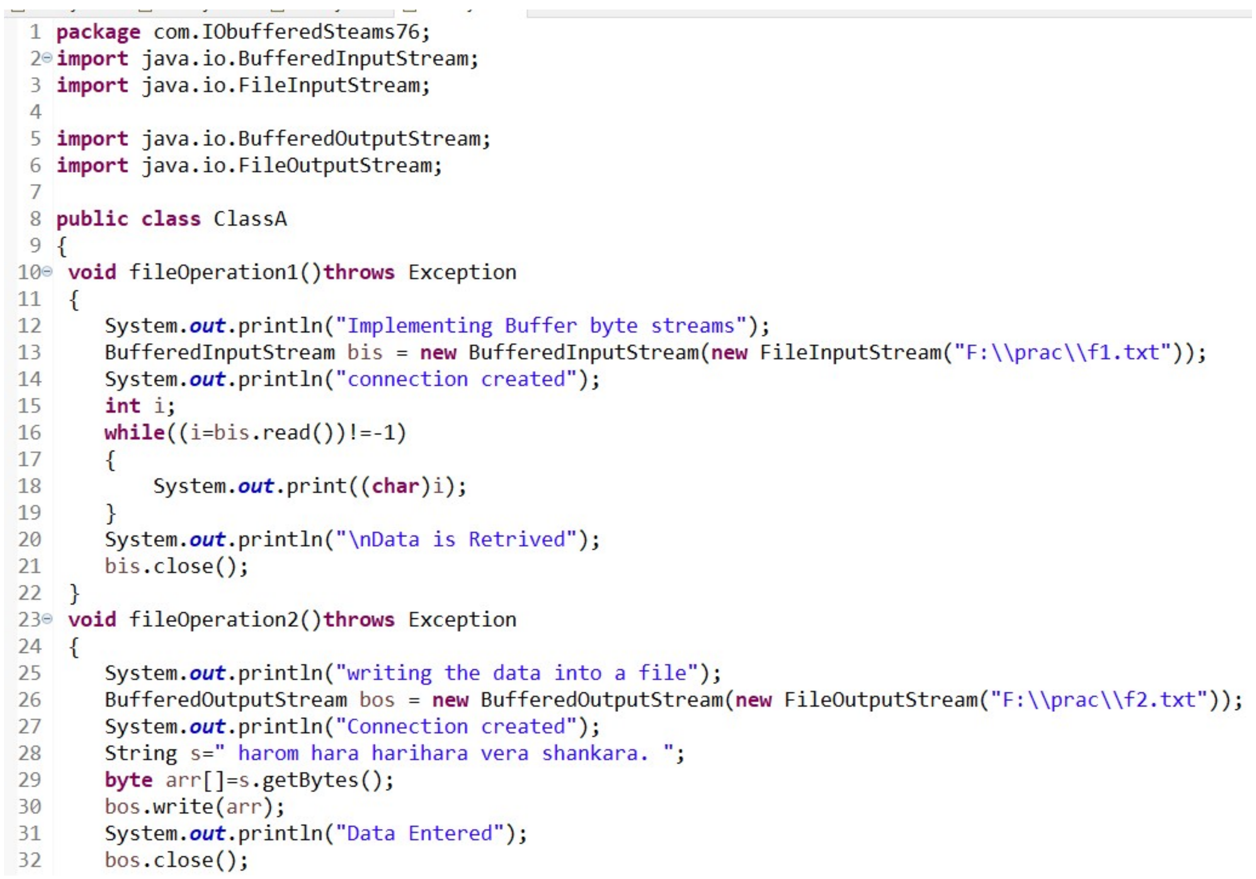


```
public class ClassA
{
    void fileOperations1()throws Exception
    {
        System.out.println("Reading the data from a file by using BufferedByteStreams\n");

        BufferedInputStream bis=new BufferedInputStream(new FileInputStream("D:\\NIT\\file1.txt"));
        System.out.println("Conenction Created");
        int i;
        while((i=bis.read())!=-1)
        {
            System.out.print((char)i);
        }
        System.out.println("\nData Reterived");
        bis.close();
    }
    void fileOperations2()
    {
        // Write the logic to copy the file byusing BufferedCharacter Streams
    }

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

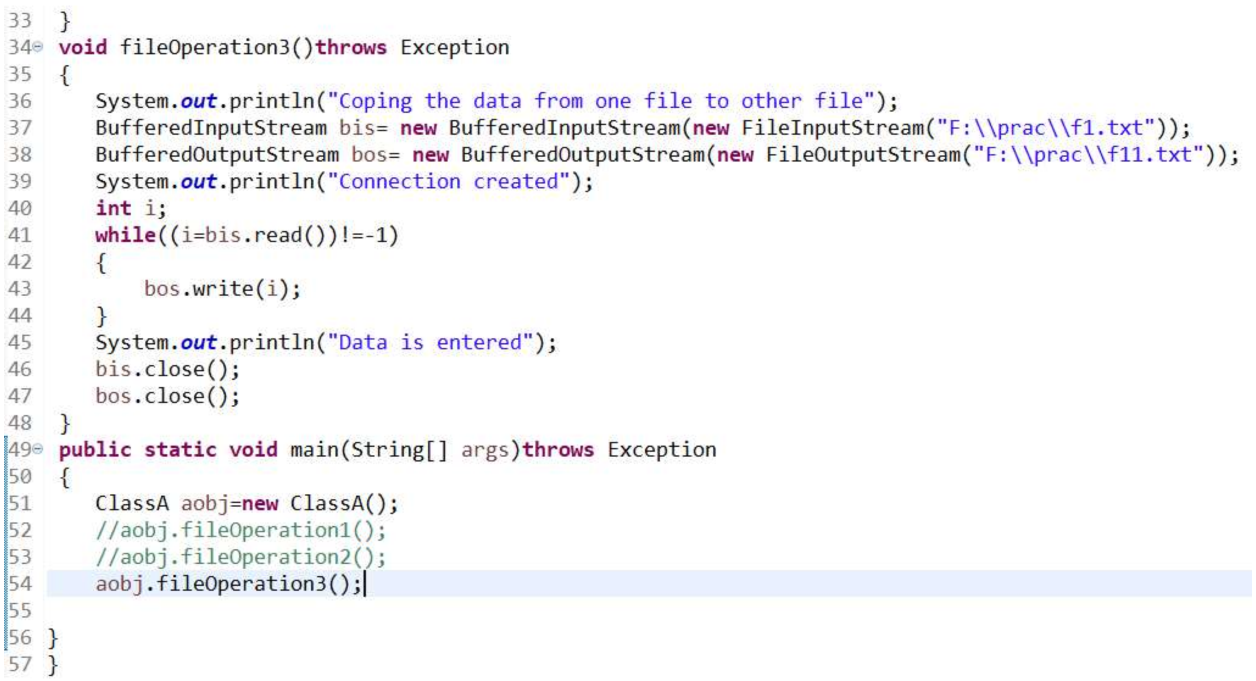


```
        ClassA aobj=new ClassA();
        aobj.fileOperations1();
    }
```

Buffered byte Stream

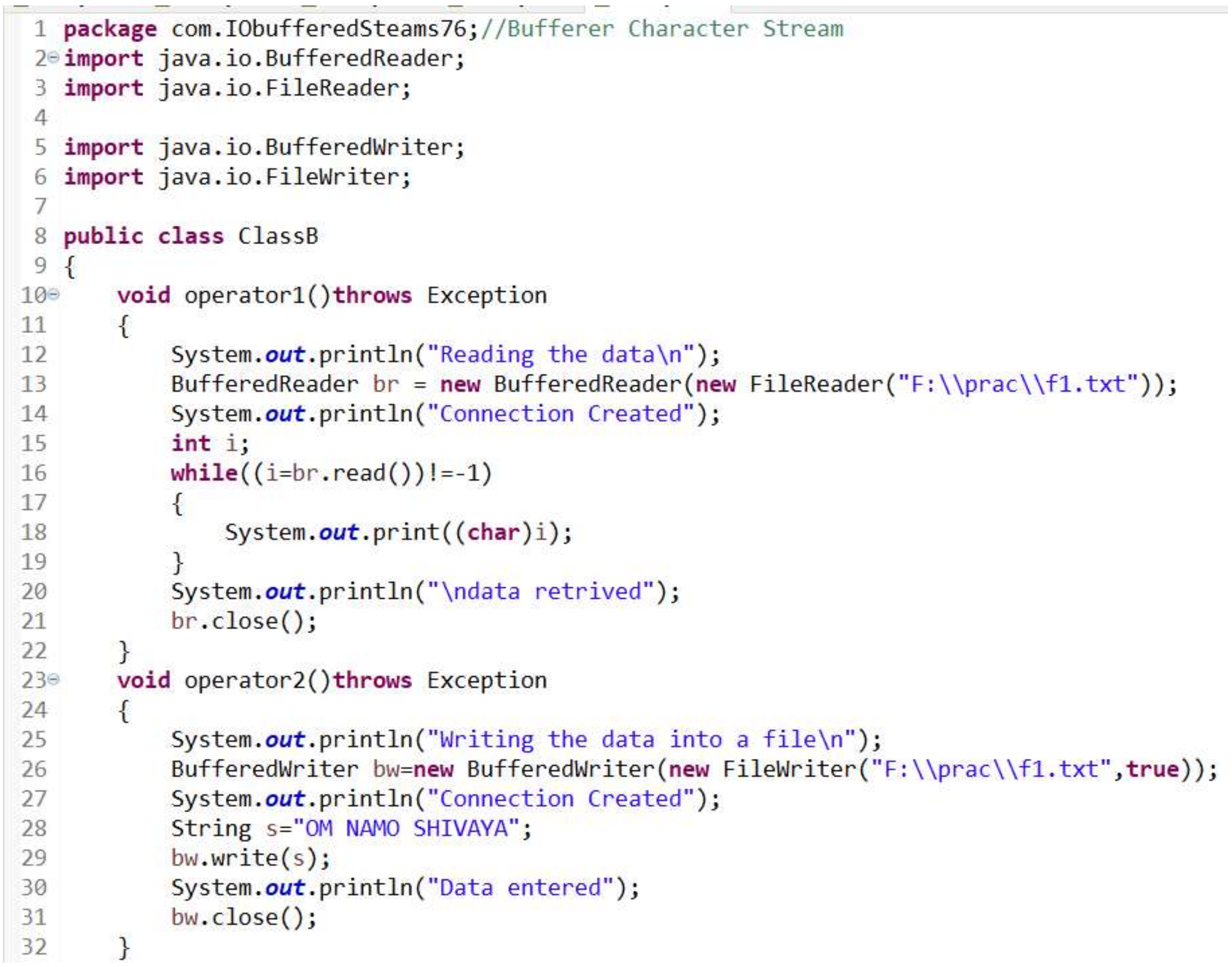


```java
package com.IObufferedSteams76;
import java.io.BufferedInputStream;
import java.io.FileInputStream;

import java.io.BufferedOutputStream;
import java.io.FileOutputStream;

public class ClassA
{
 void fileOperation1()throws Exception
 {
    System.out.println("Implementing Buffer byte streams");
    BufferedInputStream bis = new BufferedInputStream(new FileInputStream("F:\\prac\\f1.txt"));
    System.out.println("connection created");
    int i;
    while((i=bis.read())!=-1)
    {
        System.out.print((char)i);
    }
    System.out.println("\nData is Retrived");
    bis.close();
 }
 void fileOperation2()throws Exception
 {
    System.out.println("writing the data into a file");
    BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream("F:\\prac\\f2.txt"));
    System.out.println("Connection created");
    String s=" harom hara harihara vera shankara. ";
    byte arr[]=s.getBytes();
    bos.write(arr);
    System.out.println("Data Entered");
    bos.close();
```

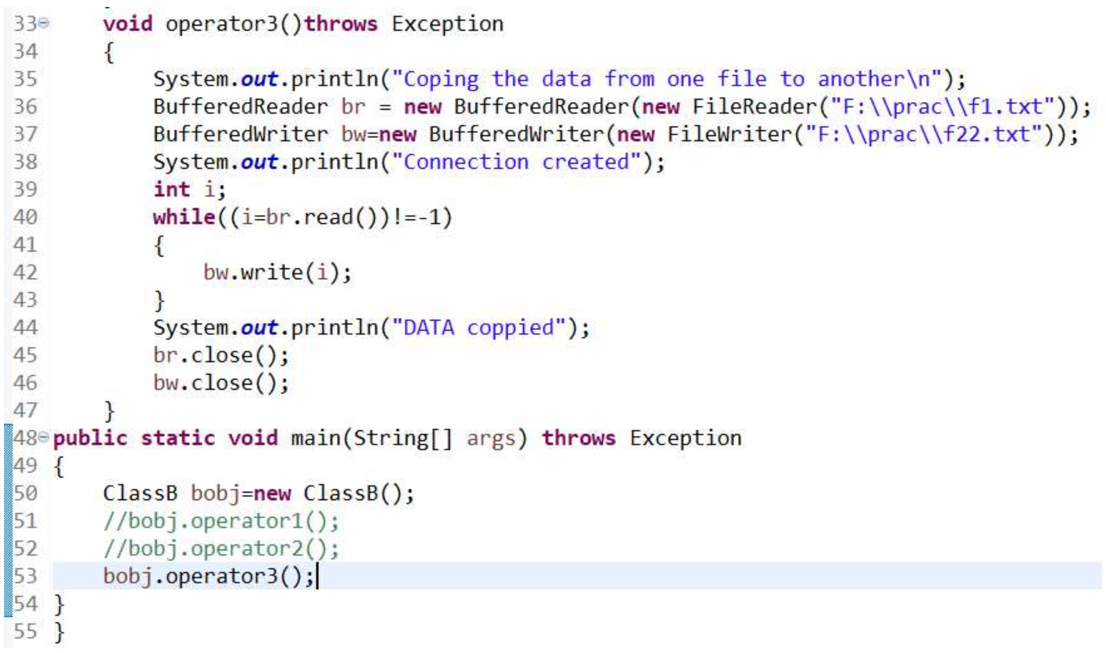


```java
 }
 void fileOperation3()throws Exception
 {
    System.out.println("Coping the data from one file to other file");
    BufferedInputStream bis= new BufferedInputStream(new FileInputStream("F:\\prac\\f1.txt"));
    BufferedOutputStream bos= new BufferedOutputStream(new FileOutputStream("F:\\prac\\f11.txt"));
    System.out.println("Connection created");
    int i;
    while((i=bis.read())!=-1)
    {
        bos.write(i);
    }
    System.out.println("Data is entered");
    bis.close();
    bos.close();
 }
 public static void main(String[] args)throws Exception
 {
    ClassA aobj=new ClassA();
    //aobj.fileOperation1();
    //aobj.fileOperation2();
    aobj.fileOperation3();

}
}
```

Buffered Character Streams

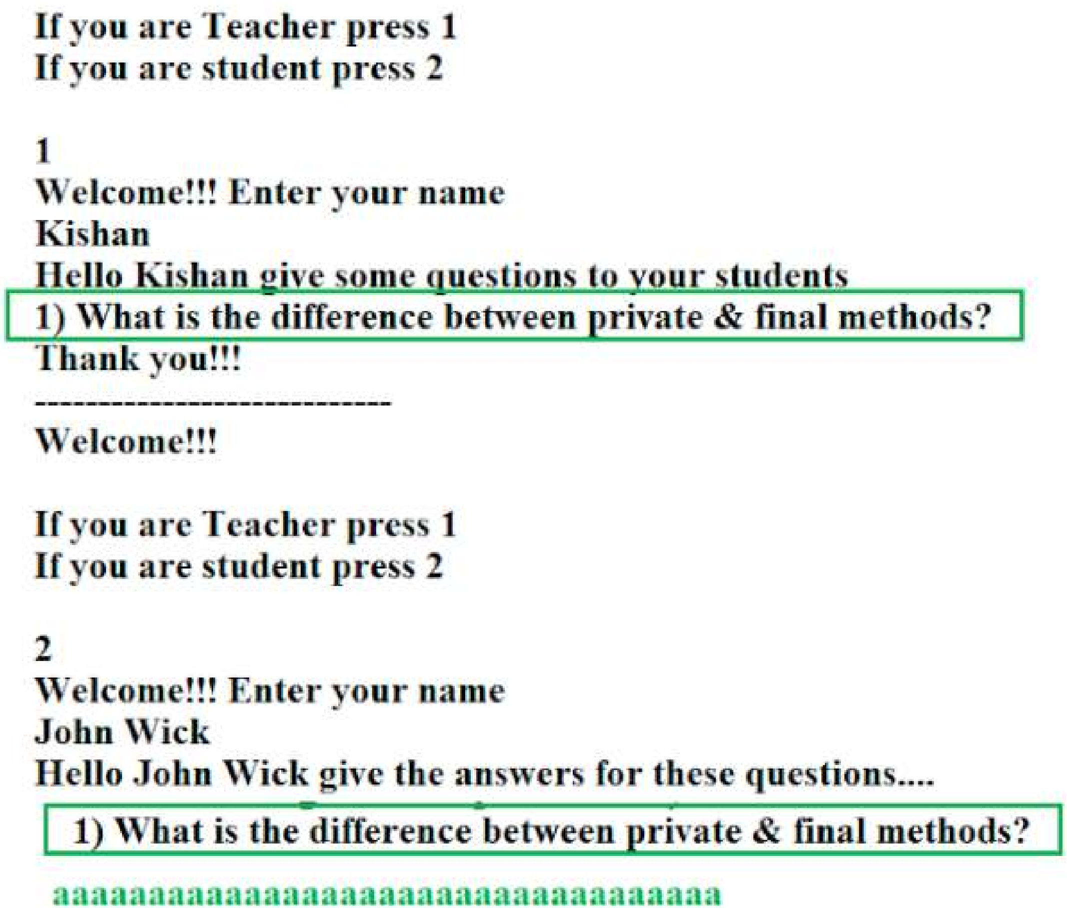


```
package com.IObufferedSteams76;//Bufferer Character Stream
import java.io.BufferedReader;
import java.io.FileReader;

import java.io.BufferedWriter;
import java.io.FileWriter;

public class ClassB
{
    void operator1()throws Exception
    {
        System.out.println("Reading the data\n");
        BufferedReader br = new BufferedReader(new FileReader("F:\\prac\\f1.txt"));
        System.out.println("Connection Created");
        int i;
        while((i=br.read())!=-1)
        {
            System.out.print((char)i);
        }
        System.out.println("\ndata retrived");
        br.close();
    }
    void operator2()throws Exception
    {
        System.out.println("Writing the data into a file\n");
        BufferedWriter bw=new BufferedWriter(new FileWriter("F:\\prac\\f1.txt",true));
        System.out.println("Connection Created");
        String s="OM NAMO SHIVAYA";
        bw.write(s);
        System.out.println("Data entered");
        bw.close();
    }
```



```java
    void operator3()throws Exception
    {
        System.out.println("Coping the data from one file to another\n");
        BufferedReader br = new BufferedReader(new FileReader("F:\\prac\\f1.txt"));
        BufferedWriter bw=new BufferedWriter(new FileWriter("F:\\prac\\f22.txt"));
        System.out.println("Connection created");
        int i;
        while((i=br.read())!=-1)
        {
            bw.write(i);
        }
        System.out.println("DATA coppied");
        br.close();
        bw.close();
    }
public static void main(String[] args) throws Exception
{
    ClassB bobj=new ClassB();
    //bobj.operator1();
    //bobj.operator2();
    bobj.operator3();
}
}
```

Task



```
If you are Teacher press 1
If you are student press 2

1
Welcome!!! Enter your name
Kishan
Hello Kishan give some questions to your students
1) What is the difference between private & final methods?
Thank you!!!
----------------------------
Welcome!!!

If you are Teacher press 1
If you are student press 2

2
Welcome!!! Enter your name
John Wick
Hello John Wick give the answers for these questions....
 1) What is the difference between private & final methods?
 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```