I/O & File Handling

Unit-3

- Streams, types,
- System.in, Scanner, BufferedReader.
- System.out, printf(), PrintWriter
- read(), ~~[illegible]~~ readLine(), write()
- Reading & writing from file
- File class
  - working with directories - creating, deleting, size, path etc.

Q) What r streams?

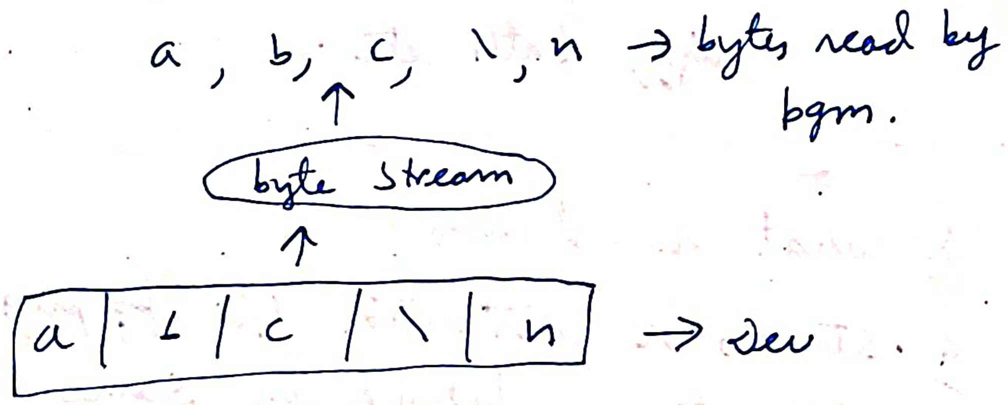
A
- Streams r abstraction b/w ~~pg~~ program & device
- All streams behave in same manner (i.e use same fun)
- So it becomes convenient to read/write using stream.
- Stream can be created to handle I/O operations on Keyboard, files etc.

Q) Types of streams?

A ① Byte stream

- reads byte from device
- No. of bytes read is same as # of bytes in dev.
- bytes r not converted to char, no translations occur.
- used for reading binary data.

eg

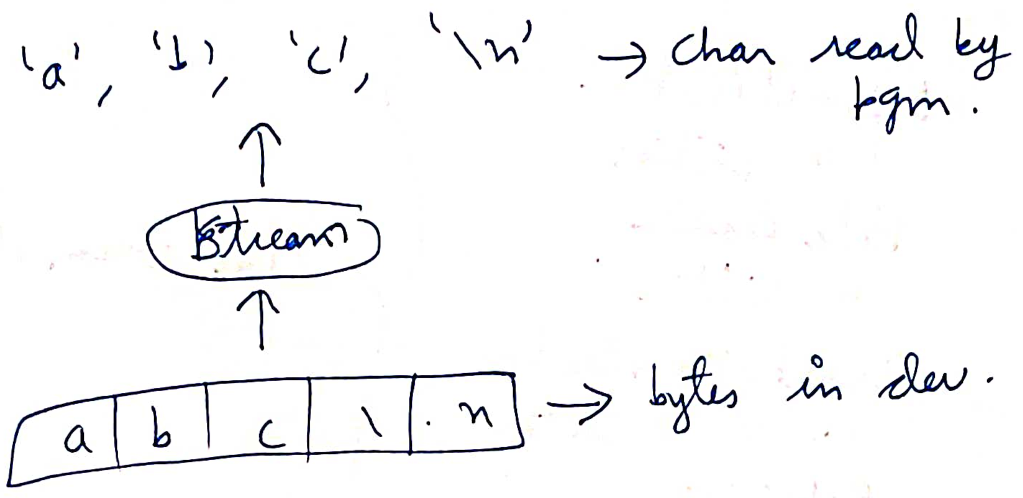


② Char stream

- reads char from dev.
- # of char read is less than # of bytes due to translations.
- bytes r converted to char, so translations occur.

eg $\underbrace{\backslash\ n}_{\text{2 bytes}} = \underbrace{\text{'\n'}}_{\text{1 cha.}}$

- Char stream preferred over byte.



Q) what r predefined streams in java

A)

$$\begin{tikzcd} & \text{java.lang.System} \arrow[dl] \arrow[d] \arrow[dr] & \\ \text{System.in} & \text{System.out} & \text{System.~~ou~~err.} \end{tikzcd}$$

$\underbrace{\text{System.in}}_{\text{I/P stream}}$ $\underbrace{\text{System.out}}_{\text{O/P}}$ $\underbrace{\text{System.err}}_{\text{err}}$

- in is an obj of ~~Print~~ InputStream class
- out is an obj of PrintStream class not System class
- PrintStream obj

Stream Classes

java.io.*;

BufferedInputStream BufferedOutputStream File Input Strea " O/P " Input Stream Output Stream. PrintStream	} Byte Streams.

$$\left.\begin{array}{l}\text{BufferedReader.}\\ \text{" Writer}\\ \text{File Reader}\\ \text{" Writer}\\ \text{InputStreamReader}\\ \text{" " Writer.}\\ \text{PrintWriter}\end{array}\right\}\ \text{Char stream class.}$$

Output Streams

- Let's see how to use 1 o/p streams

1) System.out

- It's an o/p stream.
- It provides print(), println(), printf()
- print(), println() r used for unformatted o/p.

printf() :-

- Used for formatted o/p.
- formatted o/p means we can separately specify int, char, string part of the o/p.

~~eg~~ - eg ~~Print~~ int a; string s;

System.out.printf("$\underbrace{\%d}_{\text{int}}$ $\underbrace{\%s}_{\text{string}}$ $\underbrace{\%n}_{\text{new line}}$", a, s);

• printf() ~~works~~ used in same way as in C

~~• printf()~~

• Scanner is used for formatted input.

Q) What is PrintWriter? Why is used? WAP to demo?

A) • PrintWriter is ~~to~~ used to print o/p ~~to~~.
• System.out is a byte based stream.
If Char based stream is needed then PW can be used.

eg CD{
    PSUM( ){

PrintWriter pw = new PrintWriter(System.out, true);
true means flush o/p on each line

// pw is linked to System.out
// pw converts S.out to char based stream
// use print(), println() to write

    pw.println("Hello world");
    pw.print("a" + 10);
    pw.printf("%d %s", 75, "abc");

}

}

$\boxed{\text{Input Streams}}$

- 3 ways to create Input Streams
  System.in, Scanner, BufferedReader.

1) <u>System.in</u>
   - it's a predefined stream
   - ~~char~~ byte based stream..
   - p/v (provides) 2 fun : read() & readLine()

<u>read() :-</u>
- reads "one" char & returns the ascii value
- throws IOException wh. is checked Exception so it must be handled by using throws kw
- returns -1 if EOF ~~is~~ ~~to~~ occurs..

<u>readLine():</u>
- reads a string
- throws IOException
- returns -1 if EOF found.

Q) WAP to read 10 char & 10 lines using System.in

A Class demo {

PSUM (stg ar[]) throws $\underbrace{\text{IOException}}$ {

read() & readLine() throws Checked Exception so it must be handled using throws kw.

```
//read 10 char
char c;
for(int i=0; i<10; i++){
  c=(char) System.in.read();
```

read returns an int value (ascii), so convert it char using (char)

```
  sopln (c);
}

//read 10 lines String s;
for( i=10; i<90; i++) {
    s = System.in.readline() };
    sopln(s);
  }
}
}
```

(2) Formatted i/p using Scanner :-

- Scanner is a formatter class
- Used to read formatted input
- formatted input means we can specify wh. part is int, char, string etc.
- for formatted input printf () is used as seen earlier.
- Scanner p/v these fun

  nextInt ( ) → to read int

  next Line ( ) → to read line

  next ( ). charAt (0) ; → to read char.

(3) Buffered Reader ***

Q What is BR? Why is it used? show demo?

A
- BR is an input stream
- It p/v buffered input, so advantage is it is faster than other streams.

eg How to use :-

1) Convert System.in to char based stream using Input Stream Reader

2) link Input Stream Reader to BR

$$\begin{tikzcd}
\boxed{BR} \arrow[r] & \text{buffered.} \\
\boxed{ISR} \arrow[u] \arrow[r] & \text{char} \\
\boxed{S.in} \arrow[u] \arrow[r] & \text{byte}
\end{tikzcd}$$

BR also p/v 2 fun - read(), readline()

<u>WAP to use BR:-</u>

```
CD {
  PSVM() throws IOException {
                                // read & readline() throw this checked Ex.
    BufferedReader br = new BufferedReader(new InputStreamReader
                                            (System.in));
                                // link System.in & ISR.

    Char c;
    for(i 10 times){
        c = (Char) br.read();   // reads char in ascii form.
        Soptln(c)
    }

    String p;
    for(i 1 to 10){
        p = br.readline();      // reads line

        soptln(p);
    }
  }
}
```

$\underbrace{\text{throws IOException}}$ · $\boxed{\text{read}}$ · $\boxed{\text{read line}}$

$\boxed{\text{File Handling}}$

- Reading/Writing from files

Reading/Writing involves 2 steps
1) open the file
2) R/W using read(), write()

① file is opened by creating a stream of fileinputstream, fileoutputstream
- FileNotFoundException is thrown wh. is a CheckedEx. so it must be specified after main.

② write() fun is used to write to a file
it writes 1 char at a time
- read() fun is used to read from a file
- it reads 1 char at a time

Q WAP to to write a text "This is line1 \n This is line2" to a file.
Then read this file & print the O/P ***.

A Class demo{
  psvm ( ) throws $\underbrace{\text{IOException}}_{\text{by read()}}$, $\underbrace{\text{FileNotFoundEx}}_{\text{by stream}}$ {

```
String s = " This is line1 \n This is line2";
//Write this string char by char to the file
//Create o/p stream to write to the file.
[FileOutputStream] fout = new FileOutputStream("test.txt");
                              //FNFEx can be thrown.

for(i=0; i<s.length(); i++) {
        fout.[write]( s.charAt(i)); //Writes s[i] to file.
}
fout.close();


//create i/p stream to read from a file.
[File Input Stream] fin = new FileInputStream("test.txt");
  int c;
 while( (c= fin.[read()] )!= -1) {
     //returns -1 when eof reached.
     Soplm ([(char)]c);
             c is int, contains ascii value, so convert to
                                                    char.


 }

}
}  // This is line1
      "  "  line2.
```

File class

Q) What is the purpose of File class?

A · File class doesn't operate on streams.

· deals directly with files & file system.

· used to describe properties like size, directory, path etc.

· not used for read/write to a file.

· Creating an obj of File :-

```
File f = new File("C://users/Desktop/myfolder");
```

$\underbrace{\text{C://users/Desktop}}_{\text{complete path}}$

```
File("myfolder");
```

$\underbrace{\text{myfolder}}$ relative path - i.e myfolder in current folder.

File b/w these fun :-

```
f.getPath(); //find the path of given input
f.getParent()
f.length(); //size of file or folder.
f.delete(); // folder must be empty to delete.
f.isDirectory(); // Check whether folder or not.
f.list(); // to list all files/folders in a
                                          dir.
```

Q) WAP to list all files/folders in a directory & ~~specify~~ print size?

A)
```
import java.io.*;
CD{
  PSVM(){

//open "myfolder" in current folder.
$\boxed{File}$ f = new File("myfolder");
//if it is a dir, then obtain a list of files/& folder
  in the dir
//list returns arr of string
  if(f.$\boxed{isDirectory}$()){
  String arr[] = f.$\boxed{list()}$;

  //open each file/folder in list using f'
  for(int i=0; i<arr.length; i++){
      $\boxed{File}$ f1 = new File(arr[i]);
      if(f1.$\boxed{isDirectory()}$){
          //print name, size
        Sopln(arr[i] + " is a dir");
        Sopln("size is " + f1.$\boxed{length()}$);

  }
~~if( f1. is~~
else {
    sopln( "arr[i] + " is a file"); //Print name,
    sopln ( "Size is " + f1.\boxed{length()}); size.

  }

 }
}
```

Q WAP to demonstrate creation / deletion of folder.
- create folder1 & folder2 , then delete folder1

A
```
CD{
      PSVM ( ){

  File f1 = new File ("folder 1");
  File f2 = new File ("folder 2");
   f1. mkdir ( );
   f2. mkdir ( );
   f2. delete ( );

 }

}
```