bytes ⟹ charachiters character stream classes.

readstream class ~ Inputstreamclass
write streamclass ~ opstreamclass

I-O - STREAM - II
#LECTURE - 31

USAGE OF IO SC

```
DataInput stream in = new DataInputstream (system.in);
    string a = in.readline();
    Float.ValueOf (tempstring).
    Integer.parseInt (tempstring);
```

Keyboard is identified by.

FILE HANDLING IN JAVA

file I/O.

→ File
→ fileInputStream
→ FileoutputStream
→ RandomAccessFile.

} 4 MAJOR CLASSES

```
File myF = new File (filename);          // CONSTRUCTOR - 1
File myF = new File (path, filename);    // constructor - 2
File myF = new File (myDir, filename);   // CONST - 3
```

DEALING WITH FILE NAME:

getName()   getPath()   getAbsolutePath()

getParent()   renameTo(File)

TESTING A FILE:

exists()   Canwrite()   Can Read()   isFile()   isDirectory()

isAbsolute().

GETTING FILE INFO

lastModified()
length()
delete()

DIR UTILITIES

mkDir()
mkDirs()
list()

CHECKING STATUS OF A FILE

READING A FILE

fin.read(),
fin.available()
fin.skip()

```
While (b= infile.read()!= -1){ }
    (char)b
```

WRITING ONTO A FILE

```
outfile.write(array_bytes);
```

COPY A FILE → FILE [character stream class]

```
File a = new File( . );
    ins = new FileReader(inFile);
    out = new FileWriter(outFile)

whil (ch = ins.read()) != -1) {
          outs.write(ch)
    }
```

COPY A FILE → FILE [byst SC]

```
do {
     byte b = (byte) infile.read();
     outfile.write(b);
  } while(b) = -1);
```

STORING DATA INTO A FILE:

CONCATENATION & BUFFERING

```
FileInputStream  f1 = new  FileInputStream(" ");
FileInputStream  f2 = new  FileInputStream(" ");
SequenceInputStream f3 = new  SequenceInputStream(file, file2);
BufferedInputStream inBuffer = new BufferedInputStream(file3);
BufferedOutputStream outBuffer = new BufferedOutputStream(system.out);
while ((ch = inbuffer.read()) != -1){
            outBuffer.write((char)ch);
   }
```

IO STREAM - III

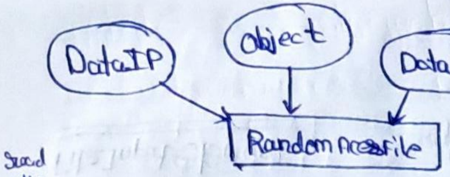
#LECTURE - 32

RANDOM ACESS FILES IN JAVA

It allows to move file pointer freely / randomly.

It allows read-write simultaneously.

RANDOM ACESSFILE allow handle files randomly in contrast to SEQUENTIAL ISC/OSC.



} can read both char/byte stream form.

r - read
w - write
rw - read & write

```
RandomAccessFile f = new RandomAccessFile ("rand.txt", "rw");
    f.seek(0);
          ↑
       Position (file ptr)

      f. writeChar(' ')
      f. writBoolean (' ');                ~ readChar();
      f. writeInt(' ');                      readInt();
                                             readDouble();
```

INTERACTIVE IP/OP:

StringTokenizer - class in utility package.

```
DataInputStream din = new DataInputStream (System.in);
StringTokenizer st;
DataOutputStream dos = new DataOutputStream(
                 new FileOutputStream(" "));

St = new StringTokenizer (din.readInt());
Int Code = Integer.ParseInt (st.nextToken());
```

GRAPHICAL IP-OP:

```
class StudentFile extends Frame {
     TextField number, name, mark;
     Button enter
     Label numLabel, nameLabel, markLabel;
     DataoutputStream dos;
     ~~resize (400,200);~~
     ~~Set~~
     Setup(){
          Resize (400,200);
          SetLayout (new GridLayout (4,2));
          number = new TextField (25);
          numLabel = new Label (" ");
          add( );
             :
          show();
     }

     addRecord (){
      dos. writeInt (new Integer (number.getText())).intValue();
      dos. writeUTF (name.getText());
     }
```

DEMONSTRATION - XII
#LECTURE-33

DataInputStream

```
DataInputStream in = new DataInputStream(system.in);
String tempstring = in.readLine();
System.out.flush();
```

FileOutputStream

```
FileOutputStream fout = new FileOutputStream();
  fout.write(65);
  fout.close();
```

Arrayoutputstream

```
byte cities[] = {'A', 'B', 'I'}
foutfile.write(cites);

String s = "Hello";
byte b[] = s.getBytes();
```

FileInputStream

"reading a text from a file"

```
try {
    FileInputStream fin = new FileInputStream("hello.txt");
    while ((int i =fin.read()) != -1) {
            sout((char)i);
    }
```

FILE status checking

```
f.getName(), f.getPath(), f.getParent(),
f.CanRead(), f.write(), f.length(), f.modified()
```

Copying $F_1 \to F_2$ using char sc:

```
File in = new File();
File out = new File();
try {
    FileReader inread = new FileReader(in);
    FileWriter outwrite = new FileWriter(out);
    int ch;
    while ((ch = inread()) != -1) {
          outwrite.write()
    }
}
```

using byte sc:

```
FileInputstream infile = null;
FileoutputStream outfile = null;
 try {
     infile = new FileInputStream ("  ");
     outfile = new FileoutPutStream ("  ");
     byte b;
      while ((b = (byte) infile.read()) != -1) {
           outfile.write(b);
     }
```

Bufferedoutput Stream

```
FileoutputStream fout = new FileoutStream (" ");
Bufferedoutputstream bout = new Bufferoutputstream(fout);
byte b[] = {'a', 'b', 'i'}
bout.write(b);
```

BufferedInputStream

```
FileInputStream fin = new FileInputStream ("");
BufferedInputstream bin = new BufferedInputStream(fin);
~~Byt~~ while ((i = bin.read()) != -1) {
           sout(i);
     }
```

SequenceInputStream (Reading a sequence of file)

```
SequenceInputstream i = new sequenceInputstream (input1, input2);
                                                 infile   infile2
```

RANDOM ACESS FILE

```
RandomAcessFile f = new RandomAcessFile "Hello.txt", "rw");
 f.writechar('x');
 f. writInt (100);
  f.writeDouble (3.1412);
   f.Seek (0);
   f.Seek (1);
    f.writeBoolean (false).
```