Java File Streams

Presented by

Sagar
Java Consultant

Data Persistence

Data is persisted in

File (I/O Streams)

Network (Emails, Web portals)

Database Managers
(SQL * Plus, MySQL, SQL Server)

File Streams

A stream that is associated with a file is a ‘File Stream’

Files provide long-term storage of large amounts of data

Files must have a name

File Class

A stream that is associated with a file is a ‘File Stream’

File class represents the files and directory pathnames

Used for creation of files and directories, file searching, file deletion, etc.

Files provide long-term storage of large amounts of data

Files must have a name

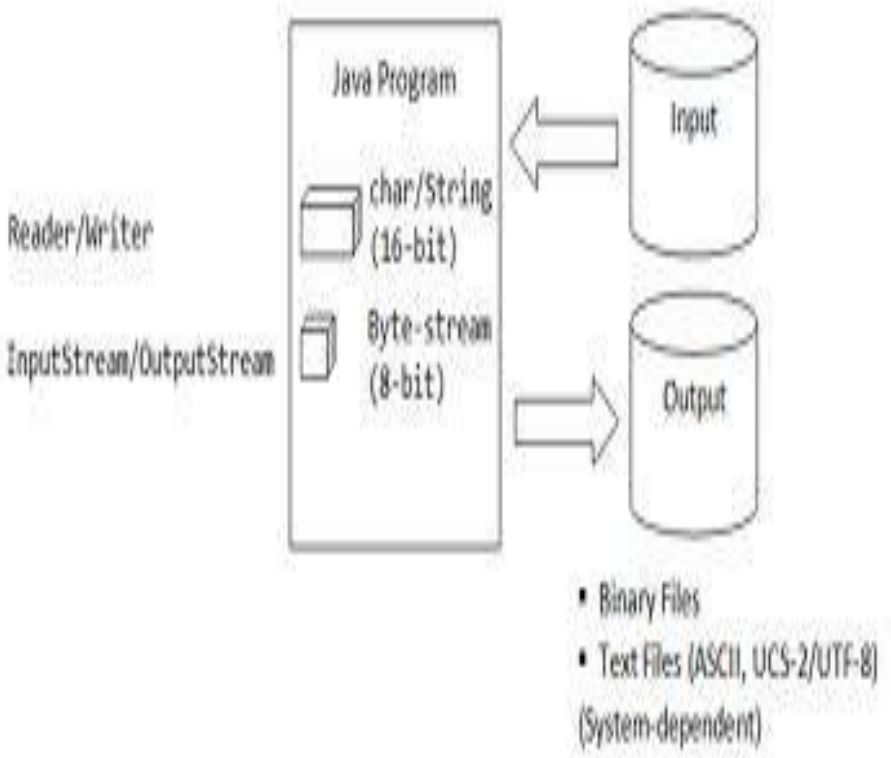
File IO Classes

FileInputStream for byte-based input from a file

FileOutputStream for byte-based output to a file

FileReader for character-based input from a file

FileWriter for character-based output a file



File IO Examples

```
FileOutputStream fw = new FileOutputStream("out.txt");
FileOutputStream fos = new FileoutputStream("out.txt“, true);

fis.write('a'); // multiple write() methods
fis.close();
```

```
FileWriter fw = new FileWriter("out.txt");
FileWriter fw = new FileWriter("out.txt", true);

fw.write('a'); // multiple write() methods
fw.close();
```

File IO Examples

```
FileInputStream fis = new FileInputStream("out.txt");
    while( (ch = (char)fin.read()) != -1) {
            System.out.println(ch);
     }
fis.close();
```

```
FileReader fw = new FileReader("out.txt");
    while( (ch = (char)fin.read()) != -1) {
            System.out.println(ch);
     }
fw.close();
```

Filtered Streams

To Write Primitive Data – DataOutputStream class

```
FileOutputStream fos = new FileOutputStream(“book.txt");

DataOutputStream dos = new DataOutputStream(fos);

dos.writeInt(1001);
dos.writeFloat(1253.25f);
```

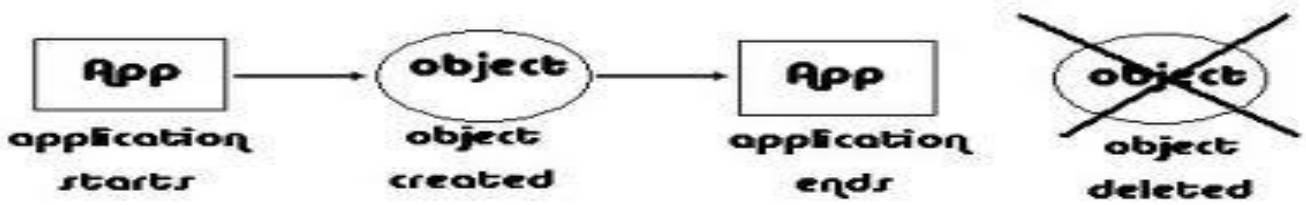
To Read Primitive Data – DataInputStream class

```
FileInputStream fos = new FileInputStream(“book.txt");

DataInputStream dos = new DataInputStream(fos);

bookId = dos.readInt();
bookId = dos.readFloat();
```

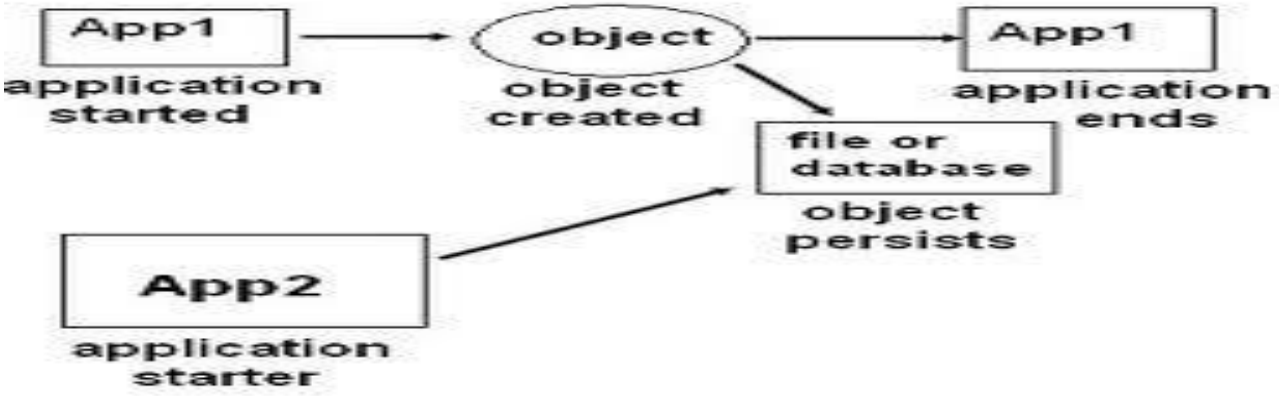
Object Serialization

Serialization is the process of persisting "object's state"

Non-Serialization



Serialization



****Object instance will be persisted instead of individual primitive types**

Serialization API

Object Serialization by - ObjectOutputStream

State is persisted by - writeObject()

Object Deserialization by - ObjectInputStream

State is persisted by - readObject()

Serialization and Deserialization Example

Serialization :

```
Employee nesha = new Employee();
  //// invoke setters  to assign data

FileOutputStream   fileOut        =    new FileOutputStream( “emp.ser” );
ObjectOutputStream objectOut   =    new ObjectOutputStream( fileOut );
   oos.writeObject( nesha);
   oos.close();
```

Deserialization :

```
FileInputStream fis                =    new FileInputStream( “emp.ser” );
ObjectInputStream ois              =    new ObjectInputStream( fis);
Emp emp  =  (Emp) ois.readObject();
ois.close();

 // invoke getters to show data
```

Transient Kewword

transient : **Instance variables will not be serialized**

Ex:

```
public class Employee {
        int aadharNo;
        String name;
        transient float salary;
}
```

PrintWriter class

Java PrintWriter class is the implementation of Writer class.

It is used to print the formatted representation of objects to the text-output stream.

write() method : to write on screen and also onto files

****System.out.prinln() writes result only on console**

PrintWriter Example

```
PrintWriter writer = new PrintWriter(System.out); // console

writer.write("Success is – people searching for you on Google but not on facebook!");


 PrintWriter technology  =   new PrintWriter( new File("D:\\tech.txt") );

 technology.write("Java, Spring, Hibernate, JSF,Android, PHP");

 technology.flush();
```


return
THANK
YOU
control