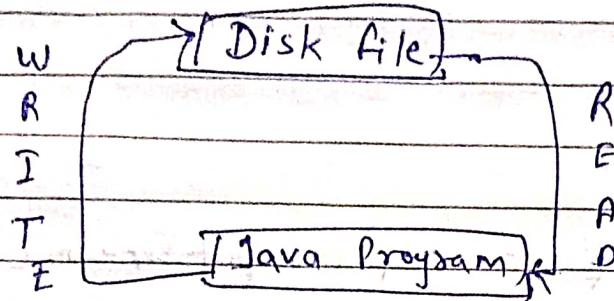


FILE HANDLING

A java program can read or write data from file called data file handling.

Java use the concept of stream to flow data.
A Stream is a sequence of bytes.



WAP that read a paragraph and save in a abc.txt.

```

import java.util.*;
import java.io.*;

class T
{
    public static void main (String [] args)
    {
        Scanner ob = new Scanner (System.in);
        try
        {
            System.out.println ("Enter a paragraph");
            String p = ob.nextLine ();
            FileWriter fw = new FileWriter ("d:\\abc.txt");
            fw.write (p);
            fw.close ();
        }
    }
}

```

d:\\abc.txt

Catch (Exception obj)

```

    {
        System.out.println("obj");
    }
}
```

OR

```
FileOutputStream fos = new FileOutputStream("d://abc.txt");
```

```
fos.write(p.getBytes());
```

```
fos.close();
```

WAP that read data from file.

```
import java.util.*;
```

```
class A
```

```
{
    public static void main (String k[])
}
```

throws Exception

```
FileReader fr = new FileReader ("d://abc.txt");
```

```
int i;
```

```
while (i = fr.read ())
```

```
while ((i = fr.read ()) != -1)
```

```
{
    System.out.println((char) i);
}
```

```
fr.close ();
```

OR

```
FileInputStream fs = new FileInputStream("d://abc.txt");
int i;
while ((i = fs.read()) != -1) // convert 2 string
    System.out.println((char)i);
}
fs.close();
```

WAP that Copy Content of abc.txt to xyz.txt.

```
try {
    FileReader fr = new FileReader("d://abc.txt");
    FileWriter fw = new FileWriter("d://xyz.txt");
    int i;
    while ((i = fr.read()) != -1) // read 2 string
        fw.write(i); // write 2 string
    fr.close(); // close 2 file
    fw.close();
}
```

java.io package :- It contain predefine class
to read or write from a file
or other resource.

class of `java.io` package :-

`InputStream` :- `InputStream` is an abstract class. It is the super class for all byte oriented `InputStream` class.

Method of `InputStream` Class :- `read()`
`close()`
`available()`

`OutputStream` class :- It is an abstract class. It is the super class for all byte oriented `OutputStream` classes.

Method of `OutputStream` Class :- `write(int)`
`write(byte[])`
`close()`
(forcefully point) \Rightarrow `flush()`

`FileInputStream` class :- It is used to read data from file. It extends from `InputStream` class. It is a byte oriented class.

`FileOutputStream` class :- It is used to write byte oriented data in file. It extends from `OutputStream` class.

`FileReader` class :- It is a character oriented class. It is used to read textual information from a file.

FileWriter :- It is a character oriented class.
It is used to write textual information in a file.

BufferInputStream Class :- It has external buffer. It is used to increase the performance of FileInputStream class.

Q) FileInputStream f = new FileInputStream ("abc.txt");
BufferedInputStream bs = new BufferedInputStream (f);
int i;
while ((i = bs.read ()) != -1)
{
 System.out.print ((char) i);
}
f.close ();

BufferOutputStream Class :- It has external buffer.
It is used to increase the performance of FileOutputStream class.

Q) FileOutputStream f = new FileOutputStream ("abc.txt");
BufferedOutputStream bs = new BufferedOutputStream (f);
String p = "Communication is a process";
bs.write (p.getBytes ());
f.close ();

Sequence Input Stream class :- qt is used to read data from more than one file.

Ex File Input Stream $f_1 = \text{new fileInputStream ("abc.txt")}$,
 file Input Stream $f_2 = \text{new fileInputStream ("xyz.txt")}$,
 Sequence Input Stream $sq = \text{new sequenceInputStream } (f_1, f_2)$;

```
int i;
while ((i = sq.read ()) != -1)
{
    sop ((char) i);
}
f1.close ();
f2.close ();
```

PrintStream Class :- qt is used to write different data types values in a file

qt has various methods to write different types of values.

```
println()
println(int)
println(char)
println(float)
```

;

```
print()
```

```
print(int)
```

;

```
printf()
```

Eg

```
int a=10;
System.out.printf("%d", a);
```

System.out.println()

it is an object of Method of
abstract Printstream Printstream
class class class.

System.out :- Standard O/P stream

System.in :- standard I/P

System.err :- Standard error stream.

Eg

```
import java.util.*;
```

class T

```
{ public static void main (String k[]) }
```

{

int a=10;

float b=10.5f;

String c="RAHUL";

FileOutput Stream fos = new FileOutputStream ("abc.txt");

PrintStream ps = new PrintStream (fos);

ps.println(a);

ps.println(b);

ps.println(c);

fos.close();

}

}

Serialization :- It is a process by which we can write an object into byte stream.

Ex

```
import java.util.*;  
import java.io.*;
```

```
class Emp implements Serializable
```

}

```
String n, id, des;  
double sal;
```

```
Void input()
```

{ }

```
Scanner ob = new Scanner (System.in);
```

```
SOPL ("Enter Name");
```

```
n = ob.nextLine();
```

}

n = Ram

id = 10

des =

sal = 2000

```
public static void main (String k[]){}
```

throws Exception

```
Emp e = new Emp();
```

```
e.input();
```

FileOutputStream fos = new FileOutputStream("stu.txt");
ObjectOutputStream ot = new ObjectOutputStream(fos);
ot.writeObject(e);
ot.close();
fos.close();
}

Deserialization :- It is an opposite process of serialization. To Read by filestream into object is called Deserialization.

Ex class R
{
public static void main (String k[]){
Emp e = new Emp();
try {
FileInputStream f = new FileInputStream ("stu.txt");
ObjectInputStream ot = new ObjectInputStream (f);
e = (Emp) ot.readObject();
System.out.println (e.id);
System.out.println (e.desi);
System.out.println (e.h);
System.out.println (e.salary);
} catch (Exception et) {}
ot.close();
f.close();
}

Q Reading Input from keyword :- Java provide many way to read input from keyword like.

1. by using Scanner class → java.util [introduce in JDK 1.5]
2. By using DataInputStream class
By using Console class.
By using BufferedReader class.

Ex:- import java.io *;

class A

{

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

 {

 InputStreamReader dr = new InputStreamReader

(System.in);

 BufferedReader br = new BufferedReader(dr);

 String p;

 SOPL("Enter Name");

 p = br.readLine();

 SOPL("Name is " + p);

}

}

Ex2 By using `Console` o- `Console()` method of `System` class return object of `Console` class

method - `String readLine()`
`char[] readPassword()`

ex ~~import java.util.*;~~

~~import java.io.*;~~

~~class A~~

~~{~~

~~public static void main(String[])~~ throws Exception

~~{~~

~~Console c = System.console();~~

~~String p;~~

~~System.out.println("Enter Name");~~

~~p = c.readLine();~~

~~SOPL("Enter Password");~~

~~char ch[] = c.readPassword();~~

~~SOPL("Name is " + p);~~

~~SOPL("Password is ");~~ Note :- `Console()`

~~for (char cho : ch)~~

~~SOP(cho);~~

Method of `System` class

return object of

`Console` class

~~}~~

DataInputStream class :- `getIn` can be used to read
input from keyword.

`DataInputStream ds = new DataInputStream(System.in);`

`SOPL("Enter Name");`

`String p = ds.readLine();`

`SOPL("Name = " + p);`