In previous class we have started to learn — BufferedReader and BufferedWriter classes.

We have know two Constructors of BufferedWriter class:-

1. BufferedWriter bw = new BufferedWriter(Writer ob);
2. BufferedWriter bw = new BufferedWriter(Writer w, int buffersize);

Note:- BufferedWriter always communicates with any (Writer Object). eg. FileWriter, PrintWriter etc.

Methods of BufferedWriter class

1) write(int ch); ⟶ To write character (single) to file
2) write(char[] ch); ⟶ To write character array to file
3) write(String s); ⟶ To write String object
4) flush(); ⟶ To release buffer that means every character of buffer to be written.
5) close(); ⟶ To close BufferedWriter.
6) newLine(); ⟶ to place a new line or Separator.

Let us use all the functions and constructors in a single program:-

```
import java.io.*;
class BufferedWriterDemo{
public static void main(String []args) throws
                                    IOException
{
  FileWriter fw = new FileWriter("abc.txt");
  BufferedWriter bw = new BufferedWriter(fw);
  bw.write(99);
  bw.newLine();
  char[] ch = {'a','b','c','d'};
  bw.write(ch);
  bw.newLine();
```

```
bw.write(ch);
bw.newLine();
bw.write("Shersshah");
bw.newLine();
bw.write("College, Sasaram");
bw.flush();
bw.close();
}
}
```

output is nothing seen because output is written onto abc.txt file. You can open & see output.

Q. When compared with FileWriter, which method is available extra in BufferedWriter ?

Ans → newLine(); // to insert a new line character.

Note:- Whenever we are closing BufferedWriter, underlying FileWriter will be closed automatically, hence we are not required to close it explicitly.

Now Next class — BufferedReader — It is a class which is used to read character data from a file. It is the most enhanced reader. By using BufferedReader we can read character-by-character and line by line both.

Constructors:-

1. BufferedReader br = new BufferedReader(Reader r);
2. BufferedReader br = new BufferedReader(Reader r, int bufsize);

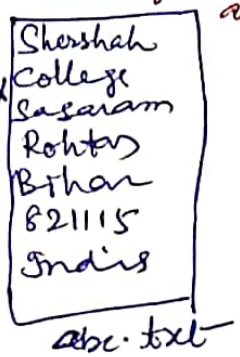
BufferedReader also does not communicate directly with File but communicates through Reader (any type) object.

Methods of BufferedReader :— Four methods are available

① int read(); ——reads character (one)
② int read(char [] ch); reads char array
③ void close(); ——closes BufferedReader

④ String readLine(); → reads line by line and returns null when finished or attempts to read next line from the file and returns null if it is not available

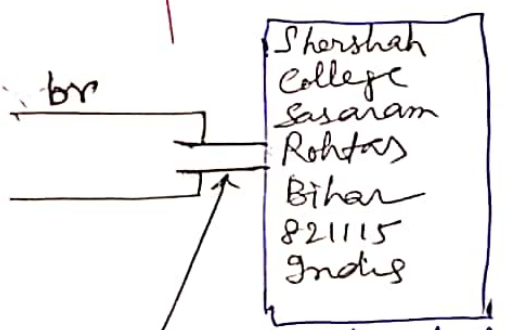
Suppose we have a file with data named abc.txt. In order to read the contents we have to use following line of codes:—



abc.txt

```
BufferedReader br = new BufferedReader(
        new FileReader("abc.txt"));
String line = br.readLine();
while ( line != null)
    {System.out.println(line);
     line = br.readLine();
    }
br.close();
```

readLine() → reads and returns a line and if not reading line available then returns null.



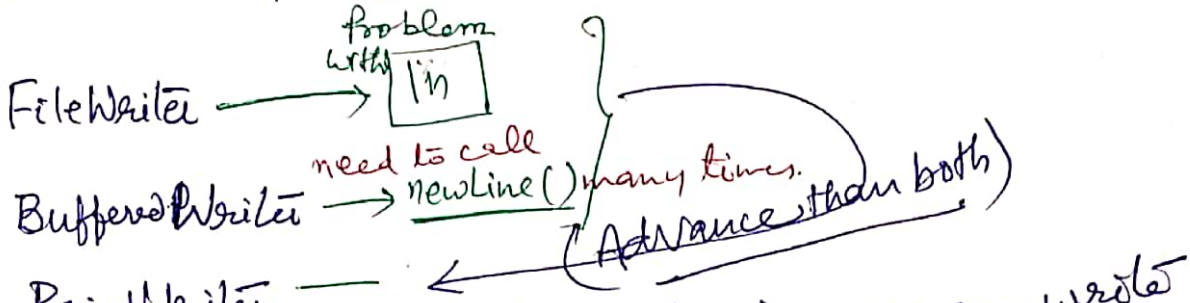
Complete program :—

```
import java.io.*;
class BRdemo{
public static void main(String args[]) throws Exception
{
   FileReader fr = new FileReader("abc.txt");
   BufferedReader br = new BufferedReader(fr);
   String line = br.readLine();
   while ( line != null)
     {
       System.out.println(line);
       line = br.readLine();
     }
   br.close();
}
}
```

new FileReader("abc.txt");

Note :- If we closed BufferedReader then the underlying FileReader will be closed automatically.

Now we will learn — PrintWriter — It is the most enhanced and powerful writer.



The main advantage with PrintWriter is — we can write any type of primitives directly to the file.

Constructors —

1. PrintWriter pw = new PrintWriter(String filename);
2. PrintWriter pw = new PrintWriter(File f);
3. PrintWriter pw = new PrintWriter(Writer w);

PrintWriter can communicate either directly to the File or via some Writer object also.

Methods of PrintWriter

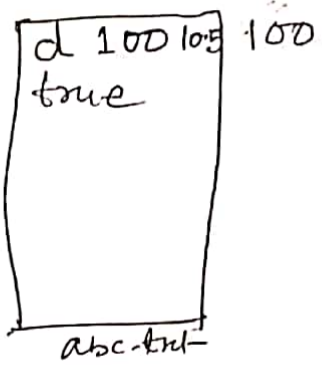
write()	print()	println()
① write(int ch);	① print(char ch);	① println(char ch);
② write(char[] ch);	② print(int i);	② println(int i);
③ write(String s);	③ print(double d);	③ println(double d);
④ flush();	④ print(boolean b);	④ println(boolean b);
⑤ close();	⑤ print(String s);	⑤ println(String s);

Difference between :— write(int ch); and print(int i);

- In case of pw.write(97); the corresponding character 'a' will be added (written) onto the File.
- But in case of pw.print(97); the int value 97 will be written onto the File.

By using PrintWriter, we can write any data type directly to the File as :-

```
File f = new File("abc.txt");
PrintWriter pw = new PrintWriter(f);
pw.print('d');      // character d
pw.print(100);      // int 100
pw.print(10.5);     // float 10.5
pw.println(100);    // int 100 with line break
pw.print(true);     // boolean true
```



⇒ The power of PrintWriter is the print() and println() methods that are previously not available with FileWriter and BufferedWriter classes.

print() and println() methods can directly write, char, int, double, boolean and String data types to the File.

Let us make use of all using a small program

PrintWriterDemo.java

```
import java.io.*;
class PrintWriterDemo{
public static void main(String args[])
                        throws IOException
{
PrintWriter pw = new PrintWriter("abc.txt");
   pw.write(105);      // write 'i' to the File
   pw.println(105);    // write 105 to the File
   pw.println(true);   // write true to the File
```

```
        pw.println('C'); //write C to the File
        pw.println("Programming");//write Programming to the File
        pw.flush();
        pw.close();
    }
}
```

Conclusion →

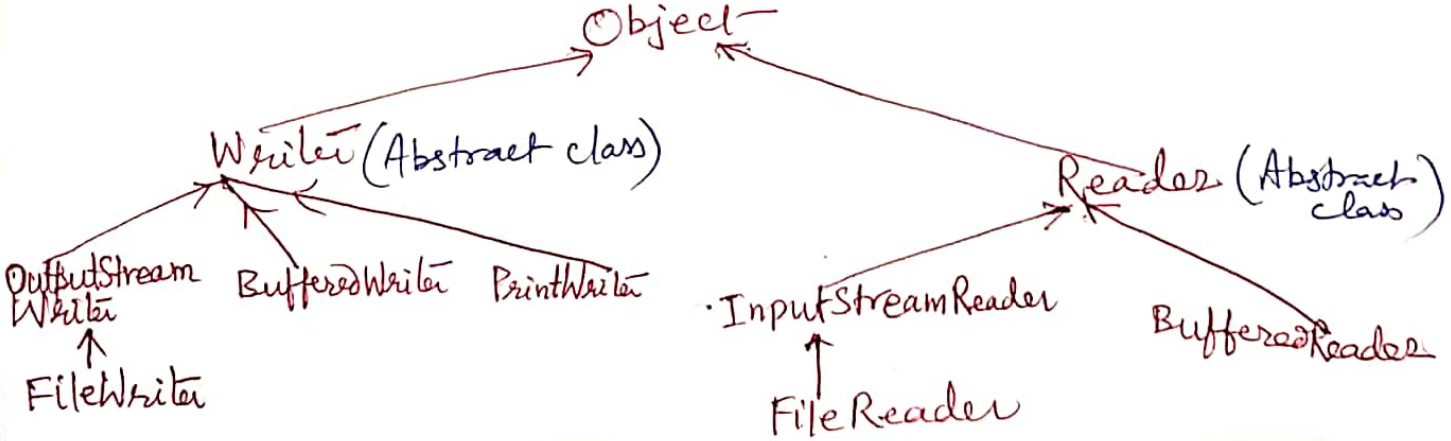
→ The most enhanced Writer to write character data to the File is PrintWriter whereas the most enhanced Reader to read character data from the File is BufferedReader.

→ In general, we can use Readers and Writers to handle character data (text-data) whereas we can use Streams to handle binary data.

We can use FileInputStream to read binary data from the File.

We can use FileOutputStream to write binary data to the File (like Image, Video, Audio files etc).

Diagram of I/O classes



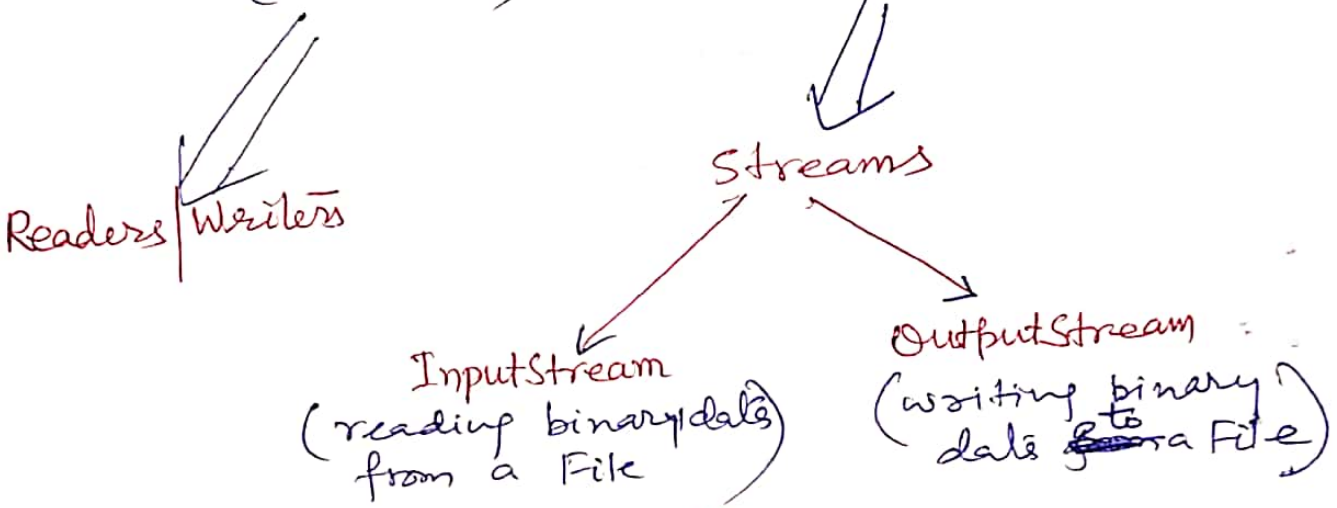
→ BufferedReader is the best choice for reading data from File

→ PrintWriter is the best choice for writing data to File.

→ In any programming language we can divide basic data into two categories —

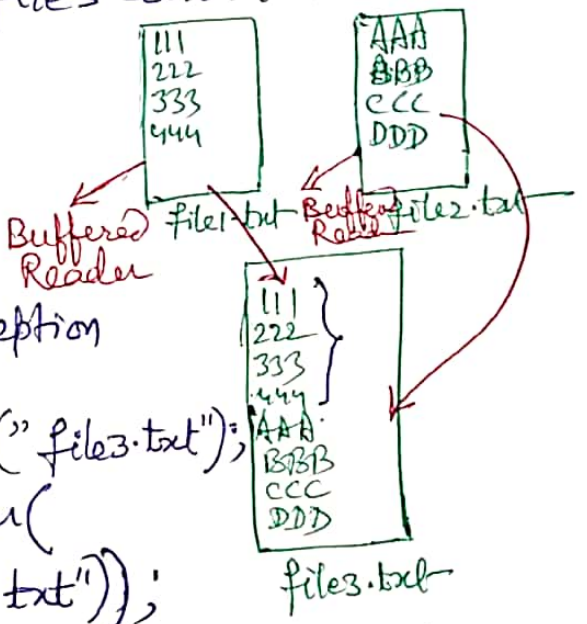
① character data (text data)

② binary data (video, audio, image etc)



Let us conclude with an example :—

Q) Write a program to merge two file's content into a third file.




```
import java.io.*;
class FileMerge
{
  public static void main(
    String [] args) throws IOException
  {
PrintWriter pw = new PrintWriter("file3.txt");
BufferedReader br = new BufferedReader(
          new FileReader("file1.txt"));
String line = br.readLine(); // got first line from file1.txt
  while(line != null)  // if it is not null
    {
```

```
        pw.println(line); //write on file3.txt
        line = br.readLine(); //go for next line
    }
    br = new BufferedReader(new FileReader("file2.txt"));
    line = br.readLine(); //read first line from file2.txt
    while(line != null) //if data found then
    {   pw.println(line); //write on file3.txt
        line = br.readLine(); //go for next line
    }
    pw.flush(); //confirms for data to be written on file3.txt
    br.close(); //close the BufferedReader
    pw.close(); //close the PrintWriter
  }
} //close of class. FileMerge
```