Simple File Input And Output

Types of Java Files

Simple File Output in Java

Simple File Input in Java

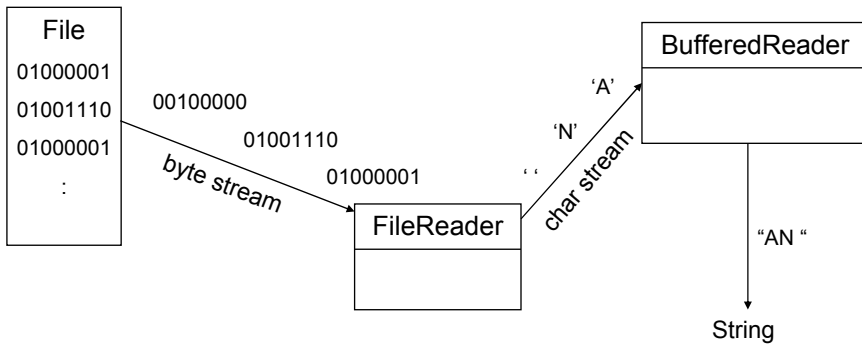# Storing Information On Files

Types of files
- Text files
- Binary files

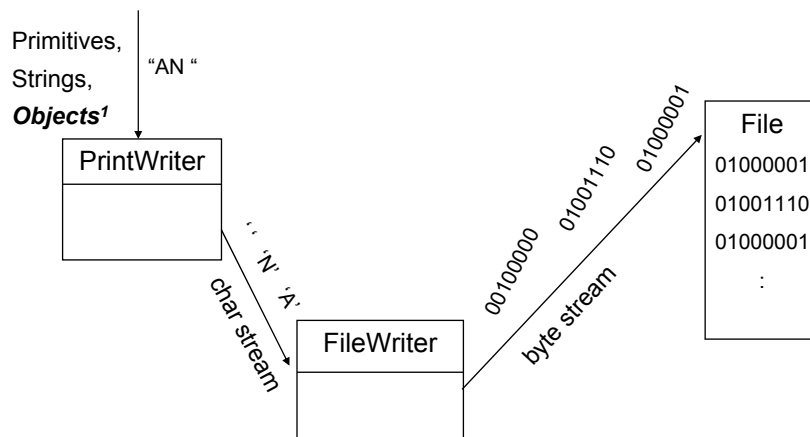# Text Files

Text files
- Every 8 bits represents a character
- e.g., '0' = 48, '1' = 49

110000	110001

'0'   '1'

Binary files
- Includes all other types of files (it's a catch all term for non-binary files)
- e.g., Interpret the bit pattern as a 16 bit unsigned short

$$110001110010_2$$

$$3186_{10}$$

(You Should Know Why You Can Get Garbage In Text Editors)

ÿØÿà   JFIF   ÿÛ C

$.' ",#   (7),01444   '9=82<.342ÿÛ C

2! !2222222222222222222222222222222222222222222222222222ÿÀ   "
        ÿÄ   ÿÄ G
     !1A   Qa"2q   '¡#3B±Á RÑáð $SbrñCs,46c'5%&t¢²ÿÄ
        ÿÄ *   !1A   "2Q a#3q B ÿÚ
        ? ò öwá»? xïMÐïäž;[¯7{ÀÀ8Û   ‚Á Žª;T   Ñ
ñ ‹ìt«›{ë^gó7E`ñ$Í¶6a´ÉB   FNOLãœU öuâ+O
    xóMÒïäžKko7z@ 'Ý ‚À$ ¬¨ÑÝ jÞ ðwŒ´Ív+ÝbõmžA$M§E ÚÑ¨d =²rËÁÇ ç
Œ n¸Œ[èÍi¾   ñ ¯§Ã}c§ù Ïæþ LÑ¤—>PÎžTlÁåÚ¨ì œŽ£ ßîG^¿²?µ?³ÿ áÿ
    øÝžt~wá÷mó¼ Þg—ÝâÙŒsœsŽø.4 cÀ¾ Óu¤Ô£:9» –
    ?â|æ¸ ¤» Ùá† Àü}x9©ëV°¯‡l—Ž-â Ck£¦›ø
K   ´Î¹_2C»â   +³œ » ÒÞx.   ™Z°<âo x'
,j/a €Ù¸¡±·¡y   ØÄ¾R±}œòvàsœ`Ô îO^ ²?µ?³ÿ áÿ øízt~wá÷mó¼ Þg—
ÝâÙŒsœs^¤|Ç¤ ý;ÏþÐÿ wþTýç÷¾o—
ß¯µeüW¨øoYÒ4æ»]Qu 3G‹K, „b&'2vÊÒ NÝ¬I@€ç Ã-Hæ,, "¥¥Ë m MOÁ>"Ò4
é¯´ÿ

Reading Text Input From A File

File
01000001
01001110
01000001
:

00100000

01001110

01000001

byte stream

FileReader

'A'

'N'

' '

char stream

BufferedReader

"AN "

String

Writing Text Output To A File

Primitives,
Strings,
Objects[1]

"AN "

PrintWriter

' ' 'N' 'A'

char stream

FileWriter

00100000 01001110 01000001

byte stream

File
01000001
01001110
01000001
:

1 By objects we of course mean references to objects

An Example Of Simple Input And Output

The full example can be found in Unix in the directory:
/home/233/examples/fileIO

Class IntegerWrapper

```
class IntegerWrapper
{
    private int num;

    public IntegerWrapper ()
    {
        num = (int) (Math.random() * 100);
    }
    public void setNum (int newValue)
    {
        num = newValue;
    }
    public int getNum ()
    {
        return num;
    }
}
```

Class SimpleIO

```java
import java.io.*;

class SimpleIO
{
    public static void main (String [] argv)
    {
        IntegerWrapper iw1 = new IntegerWrapper ();
        IntegerWrapper iw2 = new IntegerWrapper ();
        String filename = "data";
        PrintWriter pw;
        FileWriter fw;
        BufferedReader br;
        FileReader fr;
```

Class SimpleIO (2)

```java
        try
        {
            fw = new FileWriter (filename);
            pw = new PrintWriter (fw);

            System.out.println("Written to file: " + iw1.getNum());
            pw.println(iw1.getNum());
            System.out.println("Written to file: " + iw2.getNum());
            pw.println(iw2.getNum());
            pw.close();
```

Class SimpleIO (3)

```
        fr = new FileReader(filename);
        br = new BufferedReader(fr);
        System.out.println("Read from file: " + br.readLine());
        System.out.println("Read from file: " + br.readLine());
    }
```

Class SimpleIO (4)

```
    catch (IOException e)
    {
        e.printStackTrace();
    }
  }
}
```

Reading Until The End-Of-File Is Reached

```
String filename = "data";
BufferedReader br = null;
FileReader fr = null;
String temp = null;

try
{
    fr = new FileReader(filename);
    br = new BufferedReader(fr);
    temp = br.readLine ();
    while (temp != null)
    {
            :      :      :
        temp = br.readLine ();
    }
}
 :     :     :
```

Checking For More Specific Error Types

```
String filename = null;
BufferedReader br;
FileReader fr;
boolean fileError = true;

while (fileError == true)
{
  try
  {
    System.out.print("Enter name of input file: ");
    filename = Console.in.readWord();

    fr = new FileReader(filename);
    br = new BufferedReader(fr);
             :              :
    fr.close ();
    fileError = false;
  }
```

Checking For More Specific Error Types (2)

```java
 catch (FileNotFoundException e)
{
   System.out.println("File called " + filename +
                         " not in the current  directory");
}
catch (IOException e)
{
   System.out.println("General file input error occured.");
   e.printStackTrace();
}
}
```

You Should Now Know

How to write to files with Java classes
- FileWriter
- PrintWriter

How to reading text information from files with Java classes
- FileReader
- BufferedReader