File Input Output

#### Constructor ##################

FileWriter fw = new FileWriter(String filepath);

FileWriter fw = new FileWriter(File file); // file reference

int read(). It attempt to read next character from the file and return it's UNICODE value. If the next character is not available then this method return -1

As this method returns UNICODE value ( int value) , at the time of prinitng we have to perform type casting

############# Example############

FileReader fr = new FileReader("abc.text");

int i = fr.read();

while(i!=1)

{

System.out.println(char(i)); // Type casting is done here to chance int unicode value to characters

i=fr.read();

}

############################# Method

int read(char[] ch)

it attemps to read enough characters from the file into file array and returns number of characters copied from the file.

File f = new File(

char[] ch = new char[(int)fr.length()]

FileReader fr = new FileReader(f); // either file path or file reference

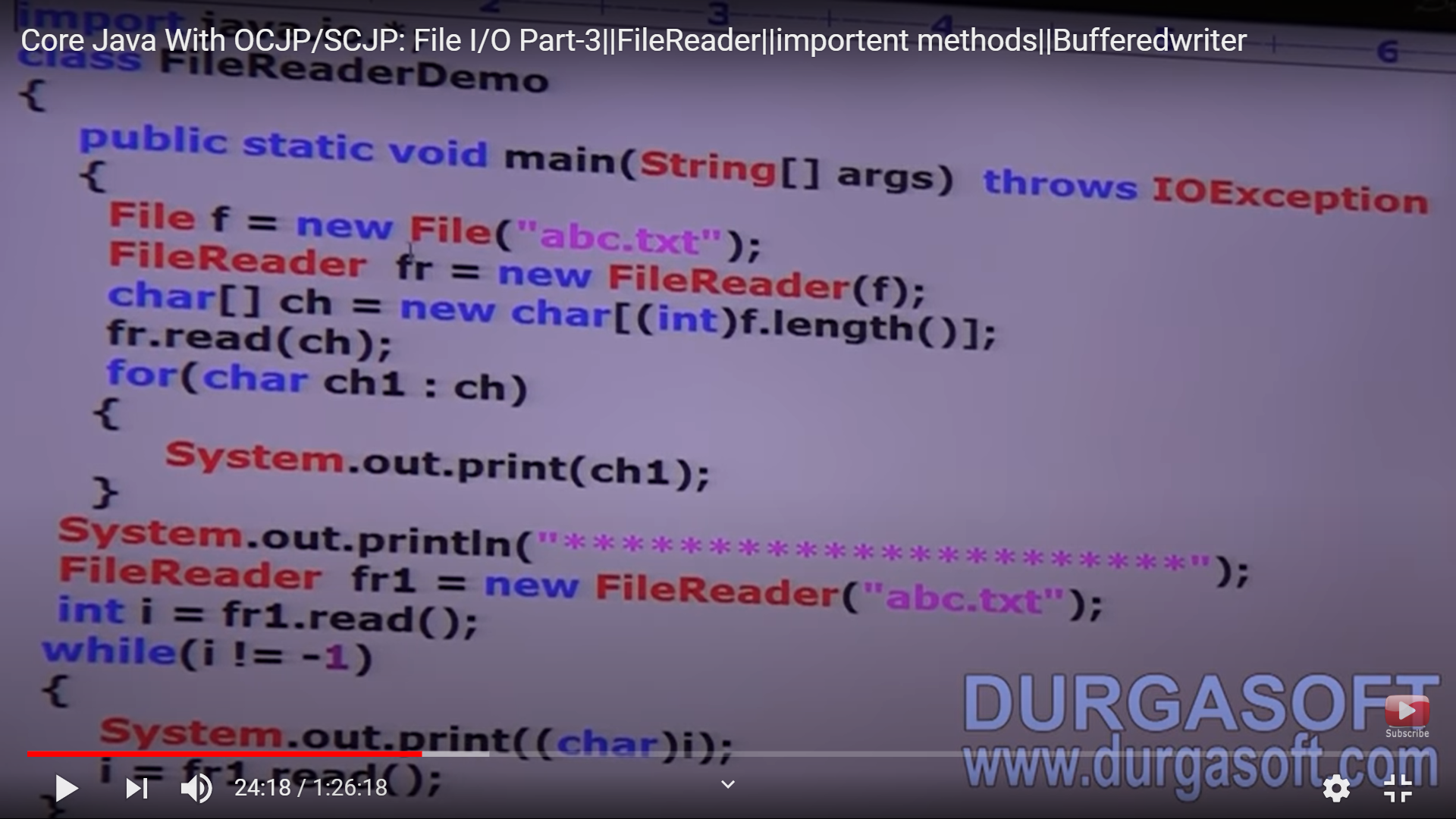
fr.read(ch) // it will read all characters in file and put into ch array character

for ( char ch1 : ch)

{

Syso(ch1);

}



In FileReader, we can read character by character which is not convenient to the programmer. It is not recommend to use.

############################ uses of FileReader and File Writer are not recommend because

1. While writing data by File Writer, we have to insert lines separator \n manually which is varied from system to system. It is difficult to the programmer.
2. By using FileReader, we can read data character by character which is not convenient to the programmer.

To overcome these problem, we should go for BufferedWriter and BufferedReader

####################### BufferedWriter#################

Constructors

BufferedWriter bw = new BufferedWriter(“bac.txt”)

##################################