Java for Beginners

Intermediate

Level 7

Mr. Teasdale

Levels of Java coding

- 1: Syntax, laws, variables, output
- 2: Input, calculations, String manipulation
- 3: Selection (IF-ELSE)
- 4: Iteration/Loops (FOR/WHILE)
- 5: Complex algorithms
- 6: Arrays
- **7: File management**
- 8: Methods
- 9: Objects and classes
- 10: Graphical user interface elements

Files

There are two types of files in computing:

- **Text files** (that contain ASCII/Unicode characters) – e.g. TXT, CSV
- **Random Access Files** (that contain binary objects) – e.g. JPG, MP3

Setting up the **file connection**

```
1  import java.io.*;
2
3  public class PlayWithFiles
4  {
5      public static void main (String[] args) throws IOException
6      {
7          //code goes here
8      }
9  }
```

Steps to remember:

1. Import: **java.io.***
2. Throw away any IOExceptions (**throws IOException**) that could potentially occur in the main method

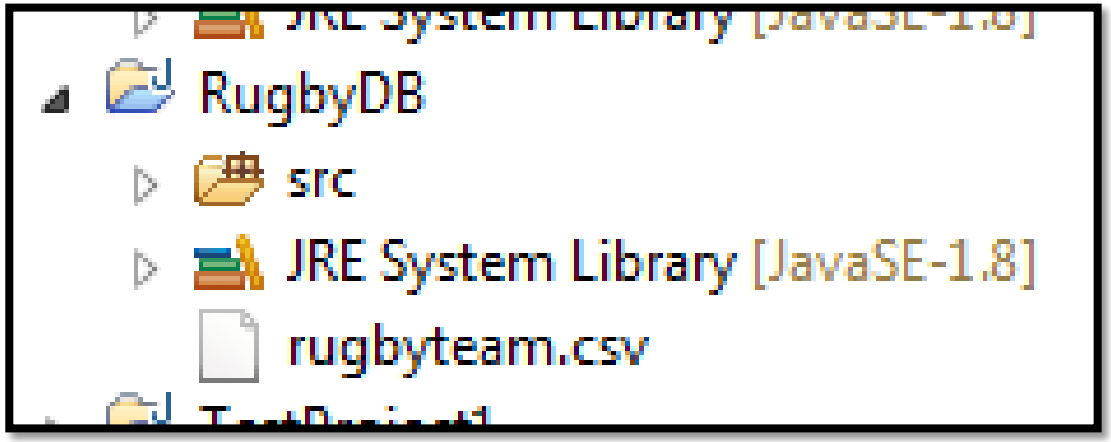
Connecting to the file

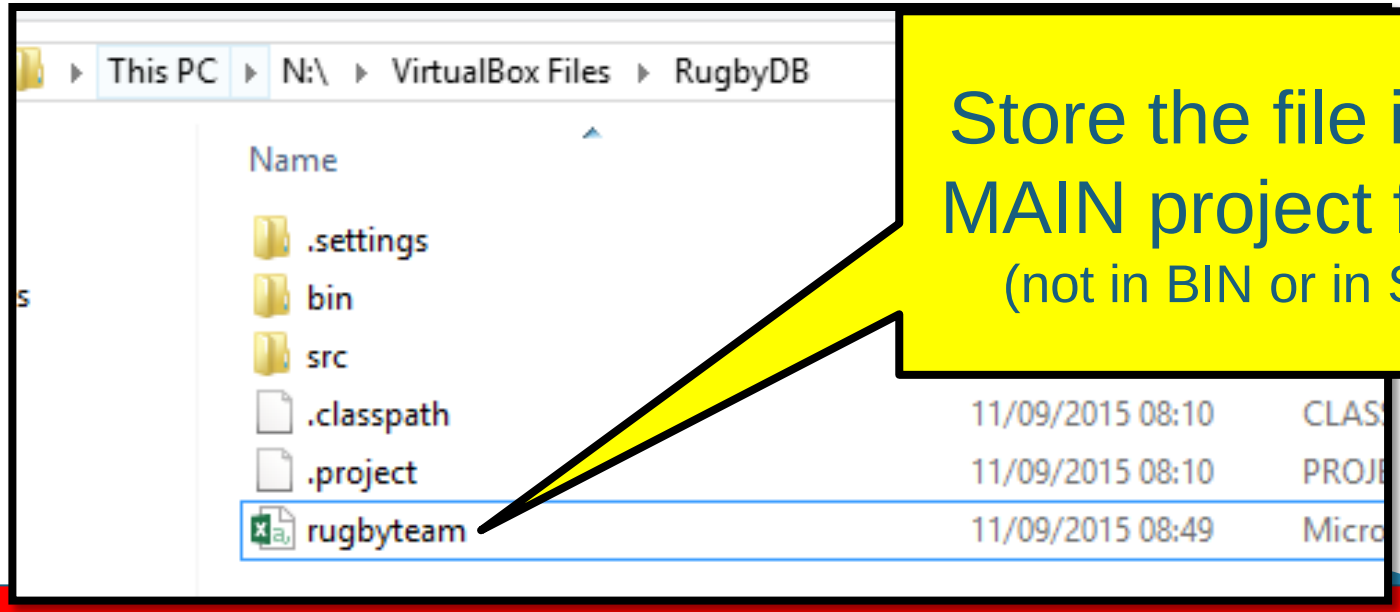
```
1  import java.io.*;
2
3  public class PlayWithFiles
4  {
5      public static void main (String[] args) throws IOException
6      {
7              //Connect Java to the file in Windows
8              FileReader fr = new FileReader("hello.txt");
9
0              //Connect a reader to the file
1              BufferedReader br = new BufferedReader(fr);
2      }
3  }
```

Steps to remember:

1. First add a File Reader
2. Then add that File Reader to a Buffered Reader

Where do I store the text file?





How do I read from the file?

```
import java.io.*;

public class PlayWithFiles
{
    public static void main (String[] args) throws IOException
    {
        //Connect Java to the file in Windows
        FileReader fr = new FileReader("hello.txt");

        //Connect a reader to the file
        BufferedReader br = new BufferedReader(fr);

        //moves the first line of text from the file
        //into a String variable
        String line = br.readLine();

    }
}
```

ALL INPUT IS ALWAYS STRING!

Useful strategy to know

Sometimes you might want to read in data like this:

```
23, 45, 56, 93, 23, 35
```

Suggested strategy:

1. Read in the line in to a String
2. Split the String into an array
3. Convert the values into ints

In Java that would be:

```
String line = br.readLine();
String[] tempstringarray = line.split(",");
int[] intarr = new int[tempstringarray.length];
for (int i = 0; i < tempstringarray.length; i++)
{
    intarr[i] = Integer.parseInt(tempstringarray[i]);
}
```

Which reads as…

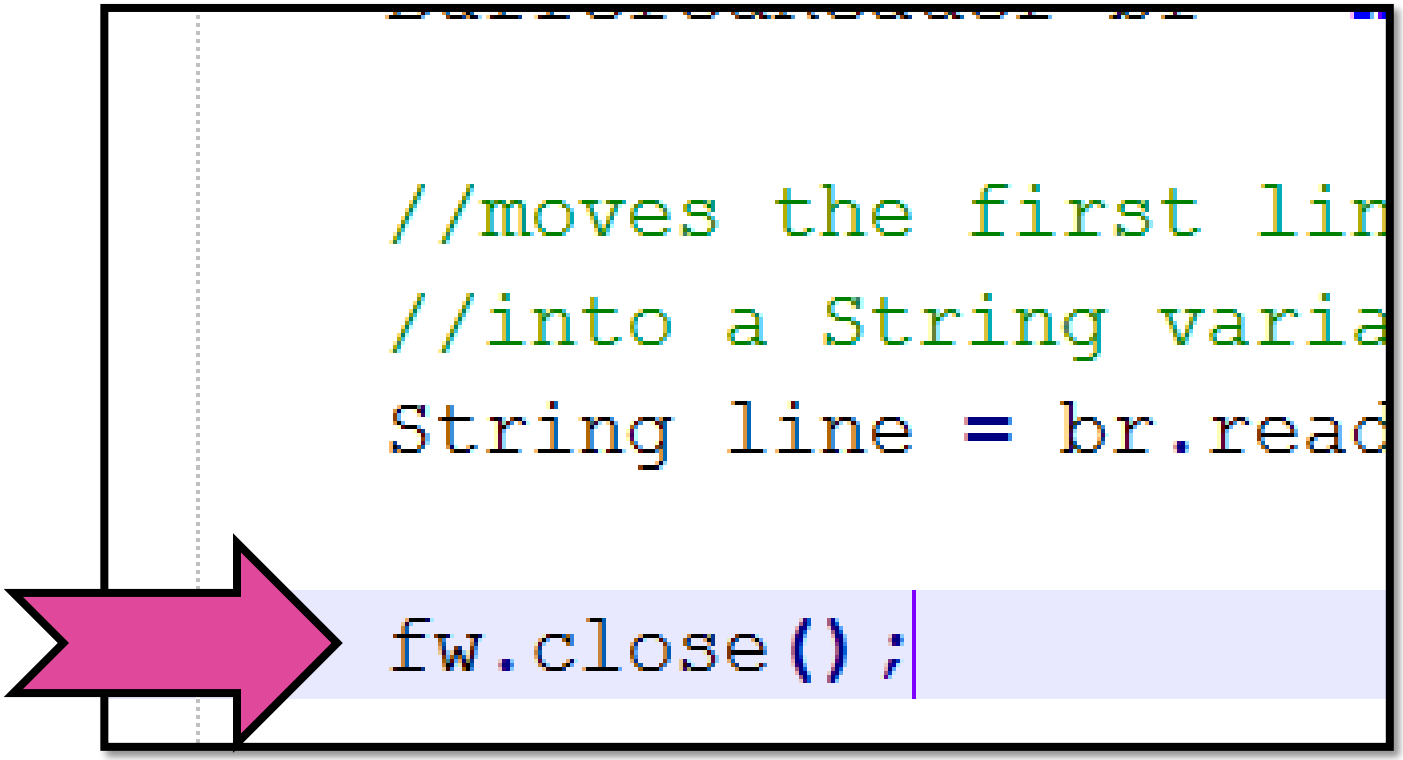
1. Read the first line from the file in to a String variable called **`line`**.
2. Split the line variable into an String array called **`tempstringarray`**
3. Create an integer array called **`intarr`** of the same size as **`tempstringarray`**
4. Loop through for all the values in **`tempstringarray`**
5. Convert each value from a String to an int and store it in the new integer array at the same index point.

Writing to a file

```
1  import java.io.*;
2
3  public class TestWriter
4  {
5      public static void main(String[] args) throws IOException
6      {
7          FileWriter fw = new FileWriter("chris.txt",true);
8          //the 'true' is whether it is appending or not
9
10         fw.write("This is a line \n");
11         // to move the file pointer to a new line, add: \n
12         // \n means <enter> in file terms
13
14
15     }//end: main
16 }//end: class
```

No buffer is used this time.
You write directly to the file.

In the end, always **.close()**



Example of adding numbers in a file

```
import java.io.*;

public class PlayWithFiles
{
    public static void main (String[] args) throws IOException
    {
            FileReader fr = new FileReader("hello.txt");
            BufferedReader br = new BufferedReader(fr);
            //file contents: 23,45,56,93,23,35
            String line = br.readLine();

            String[] tempstringarray = line.split(",");
            int[] intarr = new int[tempstringarray.length];
            int total = 0;
            for (int i = 0; i < tempstringarray.length; i++)
            {
                intarr[i] = Integer.parseInt(tempstringarray[i]);
                total = total + intarr[i];
            }
            System.out.println(total);
            //output: 275
            br.close();
            fr.close();
    }
}
```