I/O (Input/Output)

Common Forms of Input

- User-typed input
- Reading from a file
- Command-line arguments

System arguments.

Common Forms of Output

- Printing to the console
- Writing to a file

Input/Output

- Both input and output in Java work with a "stream" which accesses the buffer (memory).
- InputStream reads data from the buffer, i.e. System.in
- OutputStream writes data from the buffer into files or the console, i.e. System.out

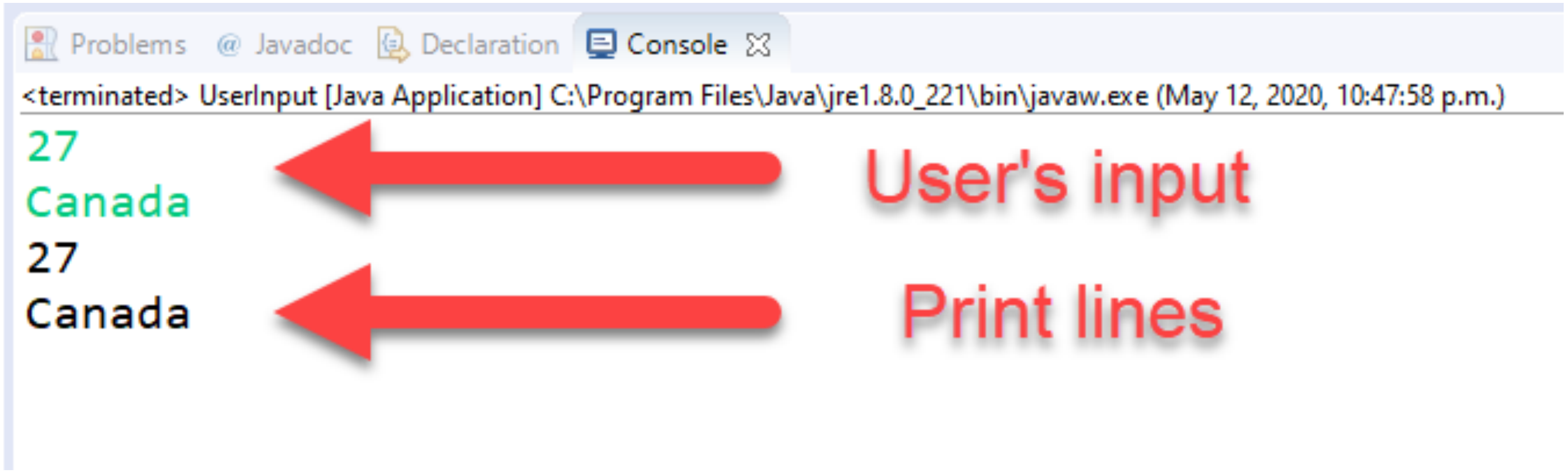
Used-Typed Input

- Java's Scanner class can help retrieve user-typed input
- It takes the InputStream as a parameter
- It then parses/converts the data stream into the specified type (i.e. int, double, etc.)   initially String. by default.

Used-Typed Input

```
Scanner sc = new Scanner(System.in);
int num = sc.nextInt();
String name = sc.next();

System.out.println(num);
System.out.println(name);

sc.close();
```

close input at the end.



```
<terminated> UserInput [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (May 12, 2020, 10:47:58 p.m.)
27
Canada
27
Canada
```

File Input

- Java's BufferedReader class is similar to Scanner but works better for file input.
- The FileReader class works closely with BufferedReader to open a file that can then be parsed and read in.
- File input could also be done with other input stream classes (as shown in ZyBooks).

File Input

- The files to read must be in the area in which the .classpath file is located.
- This will normally be the root folder, not within src or bin.

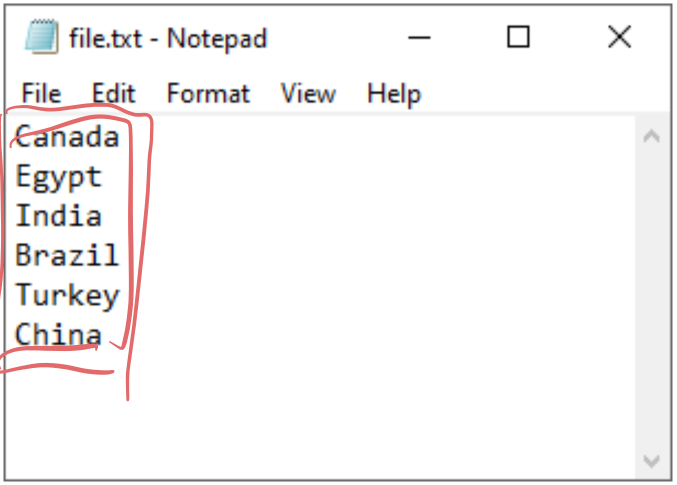
File Input

Filereader obj

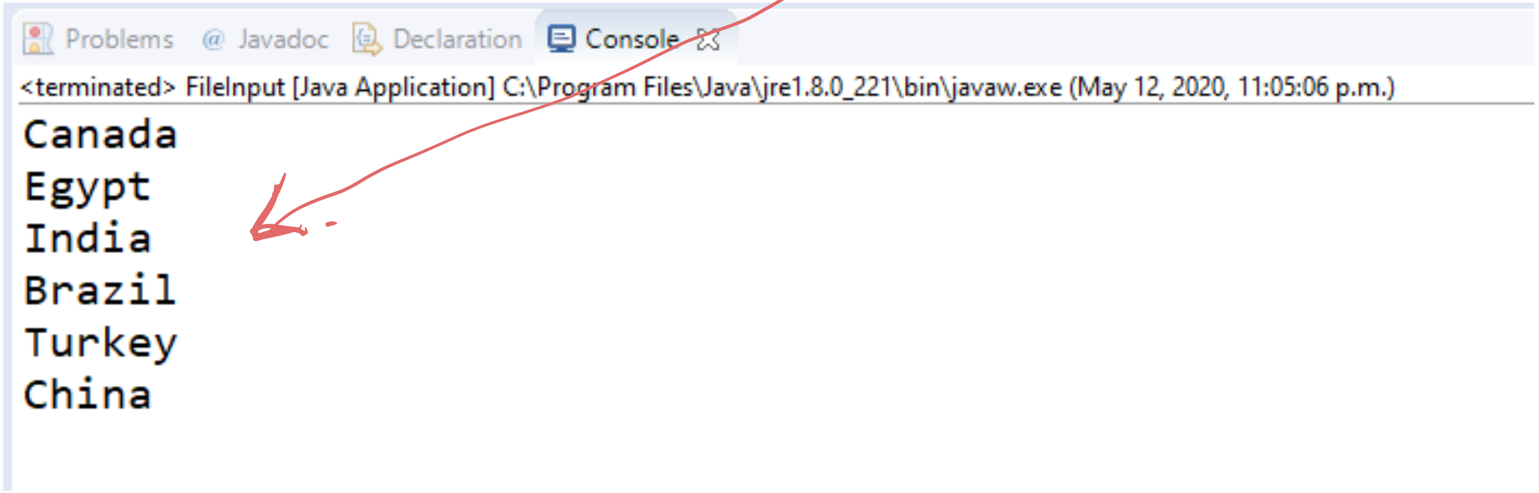
```java
BufferedReader br = new BufferedReader(new FileReader("file.txt"));

// Read first line.
String line = br.readLine();
```

read one line.

```java
// Continue reading to the end of the file.
while (line != null) {
    System.out.println(line);
    line = br.readLine();
}
```



```
Canada
Egypt
India
Brazil
Turkey
China
```



```
<terminated> FileInput [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (May 12, 2020, 11:05:06 p.m.)
Canada
Egypt
India
Brazil
Turkey
China
```

Arguments

- Have you wondered what the parameter "String[] args" in the main method is used for? sending arguments to system.
- These are called command-line arguments and are fed into the main method when you run it.
- They are useful for simulations or programs in which one of several data/map files needs to be loaded.

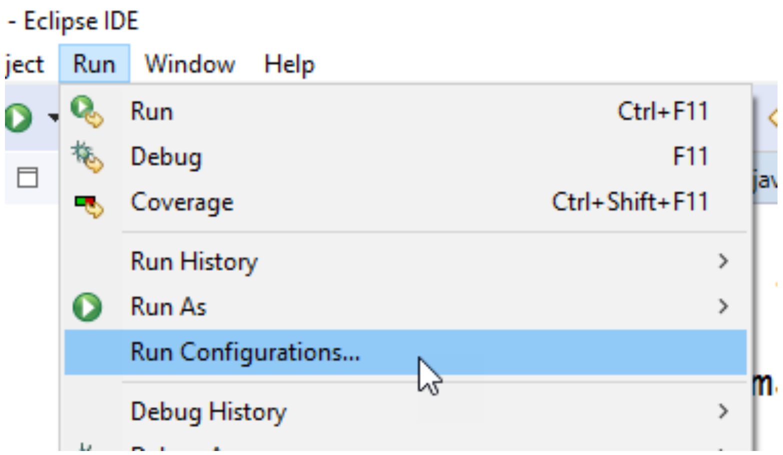
Arguments

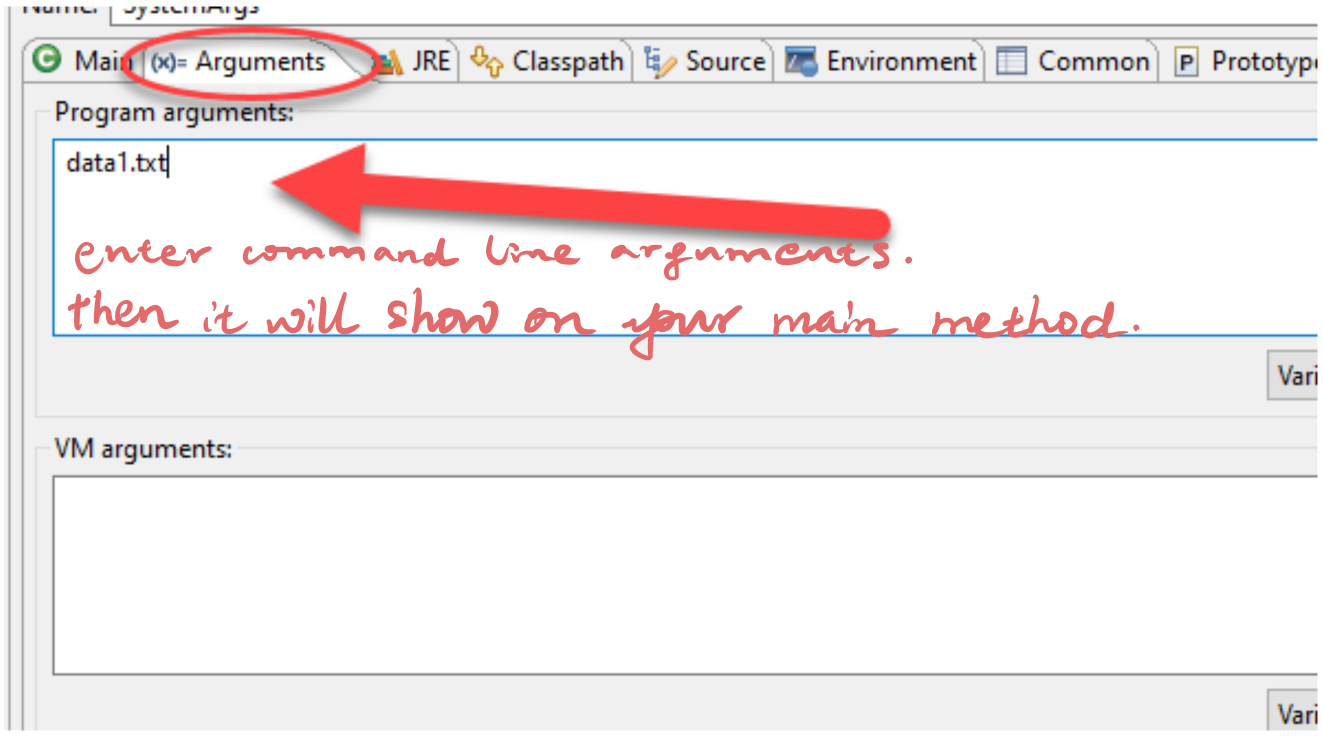
- The args come in a String array so you can have as many or few as you need.
- Use conditionals to check how many args are coming in. i.e.

-
  ```
  if (args.length == 1) {
      new MySimulation(args[0]);
  } else {
      System.out.print("No args provided");
  }
  ```

Arguments

- Where do we set these arguments in Eclipse?





data1.txt

enter command line arguments.
then it will show on your main method.

Output to Console

ends with "" ends with "\n"

- System.out.print() or .println() are used to print data to the console

Same method signature, except that the parameter list is different

- The methods are overloaded, meaning they can take in variables of any kind
  - System.out.print(192); int
  - System.out.print(45.7); float
  - System.out.print("Hello"); str

Output to Console

- Eclipse has a shortcut for this. Type "sysout" and hit Ctrl+Space. Note: I don't know if this works on all platforms (i.e. Macs) or just on Windows.

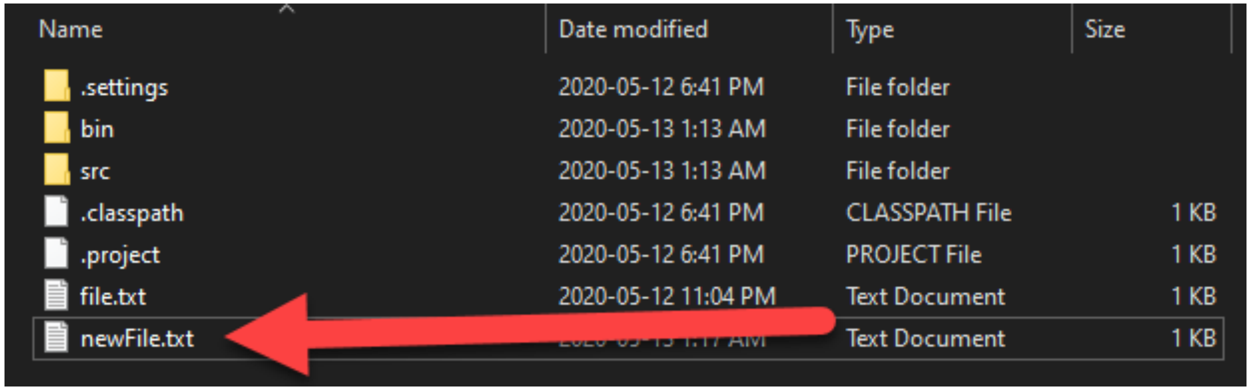
- You can also format or "pretty print" the data to the console. This topic is well-explained in the zyBooks reading (Section 9.2) so I won't cover it here.

File Output

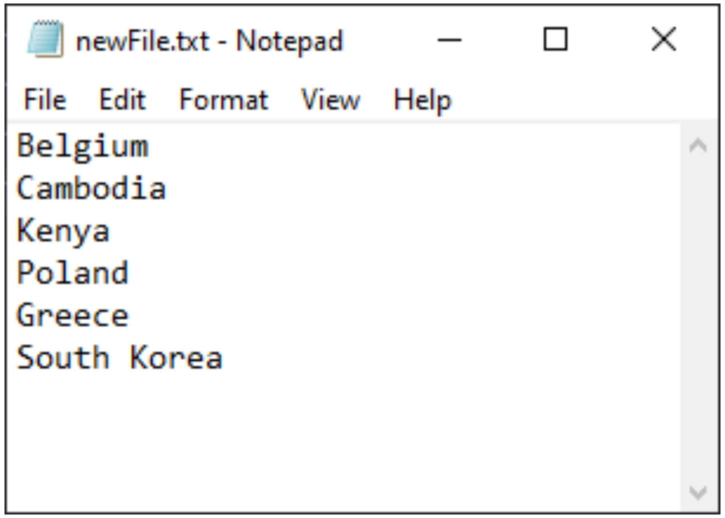
- Just like file input (reading), there are built-in data stream classes for file output (writing to a file).
- BufferedWriter and FileWriter are used together to create or open a file and write to it.
- If the file already exists, you can either write (overwrite the existing content) or append (add content to the bottom).

File Output

```
BufferedWriter bw = new BufferedWriter(new FileWriter("newFile.txt"));

bw.write("Belgium\n");
bw.write("Cambodia\n");
bw.write("Kenya\n");
bw.write("Poland\n");
bw.write("Greece\n");
bw.write("South Korea\n");

bw.close();
```



Name	Date modified	Type	Size
.settings	2020-05-12 6:41 PM	File folder	
bin	2020-05-13 1:13 AM	File folder	
src	2020-05-13 1:13 AM	File folder	
.classpath	2020-05-12 6:41 PM	CLASSPATH File	1 KB
.project	2020-05-12 6:41 PM	PROJECT File	1 KB
file.txt	2020-05-12 11:04 PM	Text Document	1 KB
newFile.txt	2020-05-13 1:17 AM	Text Document	1 KB



```
Belgium
Cambodia
Kenya
Poland
Greece
South Korea
```