(3) Store words in an array. Assume max # words = 100

```java
String[] words = new String[100];
int counter = 0;

while (filereader.hasNext())
    words[counter] = filereader.nextLine();
    counter++;
```

(4) Use `hasNextInt()`, `nextInt()` for integers

```
hasNextDouble(), `nextDouble()` for doubles
```

(5) Read combination of strings and integers.

```
deposits.txt, <SAMPLE CODE>
```

```
while (filereader.hasNext())
    String name = filereader.nextLine();
    int dep = filereader.nextInt();
    System.out.println(name + "\n" + dep);
```

Problem: nextInt() reads only an int, not entire line.

We see:

<table>
<thead>
<tr>
<th>Java sees</th>
<th>We see</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Alice \n</td>
</tr>
<tr>
<td>100</td>
<td>100 \n</td>
</tr>
</tbody>
</table>
Solution: Add `nextLine()` after `nextInt()`

```java
while (...) {
    String name = fileReader.nextLine();
    int dep = fileReader.nextInt();
    fileReader.nextLine(); // reads + discard "ln"
}
```

(Sample code): Place this code in a separate method

Writing to Files:

(1) Specify a file:

```java
String outFile = "output.txt"; // will get saved in Lecture 9/
```

(2) Create `BufferedWriter`: (other ways but this is most efficient)

```java
BufferedWriter out = new BufferedWriter(new FileWriter(outFile));
```

Special characters: `out.write("\n");` `newLine`  
`("\t");` `tab`  
`("\\")`; 

```java
// To save file, `out.close()```
```

To Do: Write a method that reads integers from `integers.txt` (see file) and writes the squares of each in another file.
filereader = new Scanner(new File("integers.txt"));

BufferedWriter out = new ... ("squares.txt");

while (filereader.hasNextLine()) {
    filereader.nextLine(); // read+ discard line I, line2...
    while (filereader.hasNextInt()) {
        int thisInt = filereader.nextInt();
        int square = thisInt * thisInt;
        out.write(square + " "); // must include " " since
        out.close();
Command Line Arguments

public static void main (String[] args)

allows the user to provide input to the program without interacting (via console) with the program.

To set

On eclipse: Run → Run Configurations → Arguments Tab
→ Program Arguments

infile.txt outfile.txt 1000

args[0]: infile.txt  
args[1]: outfile.txt  
args[2]: 1000

each stored as String

To convert integer (double) to String:

int numiters = Integer.parseInt(args[2]);