Parameter passing - pass-by-value, parameter value is copied
primitive data - value if data copied
objects - "address"

Why? Since not just a single data value

Now consider an object version of a number.

(sample code) Number class

public class Number {

    int value;

    public Number(int v) {
        value = v;
    }

}
Add to Driver:

```java
public static void swap(Number n1, Number n2)
{
    int temp = n1.value;
    n1.value = n2.value;
    n2.value = temp;
}
```

Add to main():

```java
Number numb1 = new Number(100);
Number numb2 = new Number(200);
swap(numb1, numb2);
print(numb1, value);
print(numb2, value);
```

```
OUTPUT: 200
100
swaps!
```
public static void swap2 (Number n1, Number n2) {
    int temp = n1.value;
    n1 = new Number(n2.value);
    n2 = new Number(temp);
    print(n1.value) \Rightarrow 100
    print(n2.value) \Rightarrow 200
}

main() {
    numb1.value = 200;
    numb2.value = 100;
    swap2(numb1, numb2);
    print(numb1.value) \Rightarrow 200
    print(numb2.value) \Rightarrow 100
} 

Doesn't change!
Reading/Writing Files

For file Input/Output, need:
- \texttt{import java.io.*;}
- \texttt{throws IOException - after method header of any method that performs I/O either directly or through calls to other methods.}

\texttt{ex: public static void main(...) \{ throws IOException \}

\texttt{readUserFile(),}

\texttt{public static void readUserFile() throws IOException

To read a file, need:
1. Create File Object:
   \texttt{String filename = "data.txt";} (must be stored in same directory as project)
   \texttt{File myfile = new File(filename);}
   \texttt{\textend{verbatim}}

HW3/
\texttt{src/}
\texttt{bin/}
\texttt{data.txt}
(2) Scanner to read file: (different than Scanner to read user input)

    Scanner fileReader = new Scanner (myfile);
    // when done : fileReader.close();

Combine (1) + (2):

    Scanner fileReader = new Scanner (new File (filename));

⇒ API for Scanner

* Can read a file only if there is text to read
  How to check? hasNext()

* To read a word: next()

(1) Read & print every word in the file:

    while (fileReader. hasNext())
        println (fileReader, next());

(2) Read & print every line in the file:

    while (fileReader. hasNextLine())
        println (fileReader, nextLine());

SAMPLE CODE: FileIO.java
(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.next();
```

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

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

(5) Read combination of strings and integers

`deposits.txt`, <SAMPLE CODE>

```java
while (filereader.hasNext())
    String name = filereader.nextLine(); // Alice "in" "in"
    int dep = filereader.nextInt(); // 100 "Bob"

    System.out.println(name + " " + dep);
```

Problem??

`nextInt()` reads only an int, not entire line

We see:  
Java sees:

<table>
<thead>
<tr>
<th>Alice</th>
<th>Alice &quot;in&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100 &quot;\n&quot;</td>
</tr>
</tbody>
</table>
Solution: Add `nextLine()` after `nextInt()`

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

**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 Lecture9/
   ```

2. Create `BufferedWriter`: (other ways but this is most efficient)
   ```java
   BufferedWriter out = new BufferedWriter(new FileWriter(outfile));
   ```
   ```java
   special_characters.out.write("\n"); // new line
   ("\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.