“Logical thinking and experience was as important as theory in using the computer as a tool to solve problems with programming.”

Adele Mildred Koss
Programmer of the UNIVAC I
https://courses.cs.washington.edu/courses/csep590/06au/readings/p175-gurer.pdf
import java.util.Scanner;

public class ReadInput {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);

        System.out.print("How old are you? ");
        int age = console.nextInt();
        System.out.println("You typed " + age);
    }
}

Scanner syntax

import java.util.Scanner;

public class ReadInput {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);

        System.out.print("How old are you? ");
        int age = console.nextInt();
        System.out.println("You typed " + age);
    }
}
import java.util.Scanner;

public class UserInputExample {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);

        System.out.print("How old are you? ");
        int age = console.nextInt();

        int years = 65 - age;
        System.out.println(years + " years until retirement!");
    }
}

- Console (user input underlined):
  How old are you? 29
  36 years until retirement!
The `Scanner` can read multiple values from one line.

```java
import java.util.Scanner;
public class ScannerMultiply {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);

        System.out.print("Please type two numbers: ");
        int num1 = console.nextInt();
        int num2 = console.nextInt();

        int product = num1 * num2;
        System.out.println("The product is "+ product);
    }
}
```

Output (user input underlined):

Please type two numbers: 8 6
The product is 48
Input tokens (clicker question)

- **token**: A unit of user input, as read by the *Scanner*.
  - Tokens are separated by *whitespace* (spaces, tabs, new lines).
  - How many tokens appear on the following line of input?
    23  John Smith  42.0  "Hello world"  2.50  "  19"

A. 2  B. 6  C. 7
D. 8  E. 9
When a token is the wrong type, the program crashes. (runtime error)

```java
System.out.print("What is your age? ");
int age = console.nextInt();
```

Output:

What is your age? Timmy
java.util.InputMismatchException
  at java.util.Scanner.next(Unknown Source)
  at java.util.Scanner.nextInt(Unknown Source)
  ...
```
### Scanner methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nextInt()</code></td>
<td>reads an int from the user and returns it</td>
</tr>
<tr>
<td><code>nextDouble()</code></td>
<td>reads a double from the user</td>
</tr>
<tr>
<td><code>nextLine()</code></td>
<td>reads a one-line String from the user</td>
</tr>
</tbody>
</table>

- Each method waits until the user presses Enter.
- The value typed by the user is returned.
The if/else statement

reading: 4.1
The if statement

Example:

double gpa = console.nextDouble();
if (gpa >= 2.0) {
    System.out.println("Application accepted.");
}

The *if/else* statement

Example:

```java
double gpa = console.nextDouble();
if (gpa >= 2.0) {
    System.out.println("Welcome to Mars University!");
} else {
    System.out.println("Application denied.");
}
```
Relational expressions

- **Comparison operators:**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equals</td>
<td>1 + 1 == 2</td>
<td>true</td>
</tr>
<tr>
<td>!=</td>
<td>does not equal</td>
<td>3.2 != 2.5</td>
<td>true</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>10 &lt; 5</td>
<td>false</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>10 &gt; 5</td>
<td>true</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
<td>126 &lt;= 100</td>
<td>false</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
<td>5.0 &gt;= 5.0</td>
<td>true</td>
</tr>
</tbody>
</table>
Logical operators

- Logical operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;&amp;</td>
<td>and</td>
<td>(2 == 3) &amp;&amp; (-1 &lt; 5)</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or</td>
</tr>
<tr>
<td>!</td>
<td>not</td>
<td>!(2 == 3)</td>
<td>true</td>
</tr>
</tbody>
</table>

- "Truth tables" for each, used with logical values p and q:

| p    | q    | p && q | p || q |
|------|------|--------|--------|
| true | true | true   | true   |
| true | false| false  | true   |
| false| true | false  | true   |
| false| false| false  | false  |

<table>
<thead>
<tr>
<th>p</th>
<th>!p</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
Nested if/else

Example:

```java
if (x > 0) {
    System.out.println("Positive");
} else if (x < 0) {
    System.out.println("Negative");
} else {
    System.out.println("Zero");
}
```
Exercise

- Prompt the user to enter two people's heights in inches.
  - Each person should be classified as one of the following:
    - short  (under 5'3")
    - medium(5'3" to 5'11")
    - tall    (6' or over)
  - The program should end by printing which person is taller.

Height in feet and inches: 5 7
You are medium.

Height in feet and inches: 6 1
You are tall.

Person #2 is taller than person #1.
Exercises

- Write a method that prints out if it is good weather to go for a bike ride. The weather is good if the temperature is between 40 degrees and 100 degrees inclusive unless it is raining, in which case the temperature must be between 70 degrees and 110 degrees inclusive.

- Write a method that returns the largest of three numbers using if statements.

- Write a method that determines if one day is before another day (given month and day).
Exercises

- Write a method that asks a user for 3 numbers and returns true if the numbers are all distinct.
- Write a method that determines if a number is a perfect number. A perfect number equals the sum of its integer divisors, excluding itself.
  
  6  = 1 + 2 + 3, perfect
  8 > 1 + 2 + 4, deficient
  12 < 1 + 2 + 3 + 4 + 6, excessive
Exercises

- Write a method that determines if we have time to go out for lunch. Inputs are distance to restaurant, average walking speed, time required to finish meal, time available, expected cost of meal, and money available.
- Times are expressed as whole number of minutes.
- Money is expressed as a double.