Values, Variables, and Types

Chapter 2
For Next Time

- Read Chapter 2
- Don’t forget to come to lab this week
  - Your laptops are welcome, and we can make sure Java and DrJava are properly installed and working (install via CD instead of over the ‘net)
  - Please stop by HSC 116 (the General Computing Lab) sometime before your lab meets to make sure you can log in OK
  - Be sure to read Chapter 2 before your lab meets
Numeric Values

- Integers
  - int, long, short, byte, char
- Floating point
  - float, double
- Limitations of numeric values
Nonnumeric Values

- Boolean
  - boolean

- String
  - String

- Objects
  - Programmer defined types
## The Primitive Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
<th>Range</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>byte</td>
<td>$-128 \ldots + 127$</td>
<td>8 bits</td>
</tr>
<tr>
<td>short</td>
<td>short integer</td>
<td>$-32,768 \ldots + 32,767$</td>
<td>16 bits</td>
</tr>
<tr>
<td>char</td>
<td>Unicode character</td>
<td>$0 \ldots + 65,536$</td>
<td>16 bits</td>
</tr>
<tr>
<td>int</td>
<td>integer</td>
<td>$-2,147,483,648 \ldots + 2,147,483,647$</td>
<td>32 bits</td>
</tr>
<tr>
<td>long</td>
<td>long integer</td>
<td>$-9,223,372,036,854,775,808 \ldots + 9,223,372,036,854,775,807$</td>
<td>64 bits</td>
</tr>
<tr>
<td>float</td>
<td>single-precision floating point</td>
<td>$\pm 3.4 \times 10^{\pm 38} \ldots \pm 1.4 \times 10^{-45}$ \ with at least 7 decimal digits of precision</td>
<td>32 bits</td>
</tr>
<tr>
<td>double</td>
<td>double-precision floating point</td>
<td>$\pm 1.7 \times 10^{\pm 308} \ldots \pm 4.9 \times 10^{-324}$ \ with at least 15 decimal digits of precision</td>
<td>64 bits</td>
</tr>
<tr>
<td>boolean</td>
<td>Boolean</td>
<td>false or true</td>
<td>8 bits</td>
</tr>
</tbody>
</table>
Scientific Notation

- Floating point values can be expressed in exponential form
- \(4.7 \times 10^{-5}\) is expressed as \(4.7e^{-5}\)
- Use \(e\) instead of \(E\) to improve readability
Variables and Assignment

- Variables store values
- The assignment operator (=) assigns a value to a variable
- Declarations
  - Java programs: all variables within a program must be declared
  - DrJava Interactions environment: variable declaration is optional (type deduced from the assignment)
Constants

- “Variables” that do not change
- The final specifier denotes a constant
- Compiler ensures a constant will not change
- Attempts to modify a constant results in an error
Identifiers

- Used to name variables
  - Used to name other things, also
- Must start with a letter, underscore, or $
  - Java style says don’t use underscore or $
- The rest of the name can a mixture of letters, underscores, $’s, and digits
- Java is case sensitive
- The exceptions?
Java’s Reserved Words

<table>
<thead>
<tr>
<th>abstract</th>
<th>else</th>
<th>interface</th>
<th>super</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>extends</td>
<td>long</td>
<td>switch</td>
</tr>
<tr>
<td>break</td>
<td>false</td>
<td>native</td>
<td>synchronized</td>
</tr>
<tr>
<td>byte</td>
<td>final</td>
<td>new</td>
<td>this</td>
</tr>
<tr>
<td>case</td>
<td>finally</td>
<td>null</td>
<td>throw</td>
</tr>
<tr>
<td>catch</td>
<td>float</td>
<td>package</td>
<td>throws</td>
</tr>
<tr>
<td>char</td>
<td>for</td>
<td>private</td>
<td>transient</td>
</tr>
<tr>
<td>class</td>
<td>goto</td>
<td>protected</td>
<td>true</td>
</tr>
<tr>
<td>const</td>
<td>if</td>
<td>public</td>
<td>try</td>
</tr>
<tr>
<td>continue</td>
<td>implements</td>
<td>return</td>
<td>void</td>
</tr>
<tr>
<td>default</td>
<td>import</td>
<td>short</td>
<td>volatile</td>
</tr>
<tr>
<td>do</td>
<td>instanceof</td>
<td>static</td>
<td>while</td>
</tr>
<tr>
<td>double</td>
<td>int</td>
<td>strictfp</td>
<td></td>
</tr>
</tbody>
</table>
Reserved Words

- Also called *keywords*
- You’ll find a list in Chapter 2
- You may not use a reserved word as an identifier (e.g., variable name)
- No need to memorize them
  - The compiler will let you know if you use one accidentally
Type Conversions

- \texttt{int} \rightarrow \texttt{double}, is this OK
- \texttt{double} \rightarrow \texttt{int}, is this OK?
- Use a \textit{cast} if you know it is safe to use it
  - Note: casting truncates
Arithmetic expressions