ANNOUNCEMENTS

LEARNING GOALS

• Understand at a conceptual level
  – What is Java?
  – What is DrJava?
  – What are Java primitive types and how do they work?
  – What are Java math and conditional operators?
  – What can computers do better than humans?
  – What can humans do better than computers?
MATH OPERATORS IN JAVA (+ * / - %)

- Addition
  3 + 4
- Multiplication
  3 * 4
- Division
  3 / 4
- Subtraction
  3 – 4
- Negation
  -4
- Modulo (Remainder)
  10 % 2
  11 % 2

WHAT IS DRJAVA?

- DrJava is a free Integrated Development Environment (IDE) for doing Java programming.
  - From Rice University
  - It is written in Java

MATH OPERATORS EXERCISE

- Open DrJava and do the following in the interactions pane
  - Subtract 7 from 9
  - Add 7 to 3
  - Divide 3 by 2
  - Divide 4.6 by 2
  - Multiply 5 by 10
  - Find the remainder when you divide 10 by 3
WHY IS THE RESULT OF $3 / 2 = 1$?

- Java is a **strongly typed language**.
  - Each value has a type associated with it.
  - Tells the computer how to interpret the number
    - It is an integer, floating point, letter, etc.
- The compiler determines the type if it isn’t specified (literals).
  - $3$ is an integer.
  - $3.0$ is a floating point number (has a fractional part).
- The result of an operation is the same type as the operands.
  - $3$ and $2$ are integers so the answer is an integer $1$.

CASTING

- There are other ways to solve the problem that $3 / 2$ has a result of $1$.
- You can make one of the values floating point by adding “.0”
  - $3.0 / 2$
  - $3 / 2.0$
- The result type will then be floating point.
- Or you can cast one of the values to the primitive floating point types: float or double
  - (double) $3 / 2$
  - $3 / (float) 2$

CASTING EXERCISE

- Use casting to get the values right for a temperature conversion from Fahrenheit to Celsius.
  - Celsius is $5/9 \ast$ (Fahrenheit – 32)
- Try it first with a calculator.
- Try it in DrJava without casting.
- Try it in DrJava with casting.

JAVA PRIMITIVE TYPES

- Integers (numbers without fractional parts) are represented by
  - The types: **int**, **byte**, **short** or **long**
  - 235, -2, 33992093, etc.
    - Notice that there are no commas (i.e. 14617, not 14,617).
- Floating point numbers (numbers with fractional parts) are represented by
  - The types: **double** or **float**
  - 3.233038983 -423.9, etc.
- A single character is represented by
  - The type: **char**
  - ‘a’ ‘b’ ‘A’ etc.
- True and false values are represented by
  - The type: **boolean**
  - It takes the values **true** or **false**
WHY SO MANY DIFFERENT TYPES?

- They represent different types of data.
  - Integers vs. floating point numbers vs. characters vs. boolean values.
- They take up different amounts of space.
  - How much of the computer’s memory it takes to hold one of these data types.
- They have different precisions.
  - Roughly speaking, how many decimal places of information they can hold.
    - \(3.14\) as opposed to \(3.141592654\)

SIZES OF PRIMITIVE TYPES

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>8 bits</td>
</tr>
<tr>
<td>short</td>
<td>16 bits</td>
</tr>
<tr>
<td>int</td>
<td>32 bits</td>
</tr>
<tr>
<td>long</td>
<td>64 bits</td>
</tr>
<tr>
<td>float</td>
<td>32 bits</td>
</tr>
<tr>
<td>double</td>
<td>64 bits</td>
</tr>
<tr>
<td>char</td>
<td>16 bits</td>
</tr>
</tbody>
</table>

BINARY CODES

- Inside the computer, every piece of information is either a 0 or a 1.
- These can be represented electrically or magnetically.
BINARY CODES

• A single 0/1 value is a *bit*.
  – It represents a single, simple piece of information
    • Yes/No
    • True/False
    • On/Off

• A *byte* in a computer is a group of eight bits.
  – The collection of eight bits can represent significant information.
  – If one bit can represent two things, two bits can represent four things.
    • 0 and 0, 0 and 1, 1 and 0, or 1 and 1.
  – Some number $n$ of bits can represent $2^n$ different things.
  – So, a byte can represent $2^8$ different things.
    • $2^8 = 256$.

• Another way to think of this is that a byte can represent 256 different patterns of 0s and 1s.
  • We can assign different meanings to each pattern.

  00000000
  00000001
  00000010
  00000011
  :
  :
  11111101
  11111110
  11111111