Introductory Java 3

Naming
Literals
Primitives
Java Modules

• A module is a named group of packages and related resources
• Strong encapsulation
• Explicit dependencies

```java
module java.sql {
    requires transitive java.logging;
    requires transitive java.transaction.xa;
    requires transitive java.xml;

    exports java.sql;
    exports javax.sql;

    uses java.sql.Driver;
}
```
Java Packages

Which Mary?
Mary Queen of Scots

‘Queen of Scots’ provides a namespace within which ‘Mary’ is well defined. In Java a package provides a namespace.
Java Variables

- **Instance (non-static fields, object-local)**
  - Each object has its own version (instance) of the field

- **Class (static fields, global)**
  - Exactly one version of the field exists

- **Local**
  - Temporary state, limited to execution scope of code

- **Parameters**
  - Temporary state, limited to execution scope, passed from one method to another
Java Naming

• Java names are case-sensitive
  – Whitespace not permitted
  – $, _ to be avoided
  – Java keywords and reserved words cannot be used

• Capitalization conventions
  – Class names start with capital letters (Bike)
  – Variable names start with lower case, and use upper case for subsequent words (currentGear)
  – Constant names use all caps and underscores (MAX_GEAR_RATIO)
Java’s Primitive Data Types

In addition to objects, Java has 8 special, built-in ‘primitive’ data types.

<table>
<thead>
<tr>
<th>type</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>8-bit signed 2’s complement integer</td>
<td>-128 - 127</td>
<td>0</td>
</tr>
<tr>
<td>short</td>
<td>16-bit signed 2’s complement integer</td>
<td>-32768 - 32767</td>
<td>0</td>
</tr>
<tr>
<td>int</td>
<td>32-bit signed 2’s complement integer</td>
<td>-2^{31} - 2^{31}-1</td>
<td>0</td>
</tr>
<tr>
<td>long</td>
<td>64-bit signed 2’s complement integer</td>
<td>-2^{63} - 2^{63}-1</td>
<td>0L</td>
</tr>
<tr>
<td>float</td>
<td>single precision 32-bit IEEE 754 floating point number</td>
<td></td>
<td>0.0f</td>
</tr>
<tr>
<td>double</td>
<td>double precision 64-bit IEEE 754 floating point number</td>
<td></td>
<td>0.0d</td>
</tr>
<tr>
<td>boolean</td>
<td>logically just a single bit: true or false</td>
<td>true, false</td>
<td>false</td>
</tr>
<tr>
<td>char</td>
<td>16-bit Unicode character</td>
<td>0 - 65535</td>
<td>0</td>
</tr>
</tbody>
</table>
Java Literals

• When a numerical value (e.g. ‘1’) appears, the compiler normally knows exactly what it means.

• Special cases:
  – An integer value is a `long` if it ends with ‘l’ or ‘L’
  – The prefix `0x` indicates hexadecimal, `0b` is binary:
    • `0x30` // 48 expressed in hex
    • `0b110000` // 48 expressed in binary
  – An ‘f’ indicates a float, while ‘d’ indicates double.
  – Underscores may be used to break up numbers:
    • `long` creditCardNumber = 1234_5678_9012_3456L;