A First Java Program

11 January 2008
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY
WESTERN
UNIVERSITY

What's on for today

- **Languages**: machine, assembly, high-level
- Java code **translation**
- **JDK** vs. **JRE**
- A first Java **program**
- **Comments** and **doc-comments**
- **Compiling** and **running** a Java program

TRINITY WESTERN UNIVERSITY

Review: Languages

- **Machine** language
  - "Native tongue" of computer (CPU, etc.)
  - Highly specific to machine (Pentium, Itanium..)
- **Assembly** language
  - English-like abbreviations for operations
- **High-level** language
  - More "English-like" instructions
    - Common operations: arithmetic, I/O, etc.
  - Compiler converts to machine language
- **Interpreter**: execute high-level progs w/o compile

TRINITY WESTERN UNIVERSITY

Java

- Originally for consumer electronic devices
- Then dynamic Web content (client-side)
- Now also used for
  - Large scale enterprise applications
  - Web server functionality (server-side)
  - Consumer devices (cell, Palm, etc.)

TRINITY WESTERN UNIVERSITY

Java code translation

- **Edit**: programmer writes program
  - IDE: Eclipse, NetBeans, plain-text editor, etc.
- **Compile**: compiler translates to bytecode
  - Machine-independent
- **Load**: class loader stores bytecodes in RAM
- **Verify**: check security (e.g., www)
- **Execute**: interpreter translates bytecodes into machine language

Java packages: JDK vs. JRE

- **JRE: Java Runtime Environment**
  - Everything you need to run other people's compiled Java programs
  - Interpreter translates bytecode to machine language: java
- **JDK: Java Development Kit**
  - JRE plus everything you need to write your own Java programs
  - Compiler translates Java to bytecode: javac
- On java.sun.com or Deitel textbook's CD

Java is object-oriented

- Everything is an object
  - Objects are instances of classes
- Write your program by defining classes
  - Attributes (variables; data)
  - Methods (behaviour; functions)
  - Interfaces (collections of methods)
    - A class may implement more than one interface
    - An interface may be implemented by more than one class

TRINITY WESTERN UNIVERSITY

A first Java program

- (see HelloWorld.java)
- Rule of thumb is one public class per file.
  - Same name as the *.java file
  - Sometimes can have small helper classes, too
- The main() method begins execution
  - Like C/C++
  - Declare it public and static, return type void
    - ◆ Public means other classes can see it
    - ◆ We'll get to public and other keywords later

TRINITY
WESTERN
UNIVERSITY

Comments and doc-comments

- **Comments** can either be
  - C-style: /* hi there! */
  - C++ - style: // hi there!
- **Doc-comments** start with /** (note **two** stars)
  - **Structured** comments can be interpreted by javadoc
  - Similar to Python **docstrings**
  - **@keywords**: e.g., @author, @copyright
  - **Pre/post-conditions**: @param, @return

Compile and run

- Compile: javac HelloWorld.java

- Run: java HelloWorld

- This gets a bit more complicated in a fancy IDE like Eclipse or NetBeans; see the Eclipse intro

TRINITY
WESTERN
UNIVERSITY