Outline

- Executable programs
- Assembly language
- High level languages
- Executing Programs
  - Compiled Programs
  - Java Scheme
- Translating Java Programs
Executable programs

- Computer can only execute programs written in *machine language*
- **Machine language**
  - Instructions encoded as a sequence of *bits* (0 or 1)
  - Executes *very* quickly, but impossible to read
  - Unfortunately, humans program in other languages
  - Example: 01101001 11100100 10100011
Assembly Language

- Low level language
  - One assembler instruction for each machine (binary) instruction
  - Makes programming easier, but still hard to understand
  - Simple computations may take many instructions
  - Each assembly language only works for one specific processor (Intel, Sun Sparc, Motorola, etc.)
  - Example: \texttt{mov $dec, r1}

\begin{verbatim}
  ... 8b 95 14 ff 03 95 1c ff 89 95 14 ff ...  
  mov edx, DWORD PTR _SumOfScores$[ebp]
  add edx, DWORD PTR _nextScore$[ebp]
  mov DWORD PTR _SumOfScores$[ebp], edx
  ...  
\end{verbatim}
High Level Languages

- **Are languages meant for programming**
  - *not* for executing
  - Logical and relatively English-like
  - machine independent

- **Common programming structures are part of the language**
  - each equivalent to several machine language instruction

- **Java is an example!**

Some common high-level languages:

<table>
<thead>
<tr>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBOL</td>
<td>Common Business-Oriented Language</td>
</tr>
<tr>
<td>Fortran</td>
<td>Formula Translation</td>
</tr>
<tr>
<td>Pascal</td>
<td>designed for programming instruction</td>
</tr>
<tr>
<td>Ada</td>
<td>intended for general DOD contractor use</td>
</tr>
<tr>
<td>C</td>
<td>intended for system software development</td>
</tr>
<tr>
<td>C++</td>
<td>object-oriented extension of C language</td>
</tr>
<tr>
<td>Java</td>
<td>pure object-oriented design, portable but interpreted</td>
</tr>
</tbody>
</table>
Executing Programs

- Computer processor executes each machine language instruction individually
- The program can be *compiled* or *interpreted*
Interpreted Programs

- *Interpreter* is a program that executes statements in a language
- Can think of interpreter as a (virtual) machine that executes programs in the high level language
  - That “virtual” machine can be reproduced for other processors, so the program is *portable*
  - Made from software – not hardware
- BASIC and Perl are examples of interpreted high level languages
- Generally slow… *(why?)*
Alternative is to translate programs to machine instructions

- **Compiler** – program that does translation
- Once compiled, execution is faster
- Executable (machine code) is only good for one type of processor
2. Exec Mechanics

The Java scheme:

- **Java goal is to allow compiled program to run on any computer**
  - Compiled = fast
  - Translated = portable
  - Why can’t we have both?

- **Java defines a Java Virtual Machine (VM)**

- **Machine language: Java byte code**

```
000000  ca  fe  ba  be  00 03 00  2d 00 1d 01 00 08
00000d  3c 63 6c 69 6e 69 74 3e 01 00 03 28 29
00001a  56 01 00 04 6d 61 69 6e 01 00 16 28 5b
000027  4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74
000034  72 69 6e 67 3b 29 56 01 00 03 6f 75 74
000041  01 00 15 4c 6a 76 61 2f 6c 61 6e 67 2f 50
00004e  72 69 6e 74 37 74 72 65 6d 3b 0c 00
00005b  05 00 06 01 00 10 6a 61 76 61 2f 6c 61
000068  6e 67 2f 53 74 73 74 65 6d 07 00 08 09
000075  00 09 00 07 01 00 0c 48 6e 65 6c 65 6e 5b
000082  57 6f 72 6c 66 64 21 08 00 0b 01 00 07 70
00008f  72 69 6e 74 66 64 21 08 00 0b 01 00 07 70
00009c  76 6f 6b 65 6e 67 2f 6c 61 6e 67 2f 50
0000a9  67 3b 29 56 01 00 03 6f 75 74
0000ba  61 76 61 2f 6c 61 6e 67 2f 50
0000bc  74 72 65 6d 61 6d 07 00 10 0a 00 11 00 0f
0000d0  01 00 04 43 6f 64 65 65 01 00 06 3c 69 6e
0000d4  67 3e 0c 00 14 00 02 01 00 10 6a 61
0000da  69 74 3e 0c 00 14 00 02 01 00 10 6a 61
0000e0  66 65 6e 67 2f 6c 61 6e 67 2f 50
0000e7  74 07 00 16 0a 00 17 00 15 01 00 0a 53
000104  6f 75 74 66 64 21 08 00 0b 01 00 07 70
00010b  57 6f 72 6c 66 64 21 08 00 0b 01 00 07 70
000111  5a 6f 6b 65 6e 67 2f 6c 61 6e 67 2f 50
00011e  57 6f 72 6c 66 64 21 08 00 0b 01 00 07 70
00012b  72 6c 64 2c 6a 61 76 61 2f 6c 61 6e 67 2f
000138  17 00 00 00 00 00 02 00 09 00 03 00 04
```

Programming in Java

©2002 D Barnette, B Keller & P Schoenhoff
Translating Java Programs

Java source program

compiler

compiler translates Java to byte code

Java bytecode

VM executes Java bytecode

Java Virtual Machine
Translation and File names

- Java compiler creates a file called: \textit{prog.class}
  
  ...from a file called: \textit{prog.java}

- Expects \textit{prog.java} to have
  a class in it named: \textit{prog}

\textbf{Important to remember!}