Programming

Chapter 11

Program

Computer instructions

- A recipe for solving a problem
- A set of very specific steps
  - One starting step
  - One or more ending steps
  - Next step is unambiguously determined by the current step and the data being processed
- A set of algorithms
- Typically written in a high-level language and translated into machine code (1's and 0's)

Language Levels

How to languages relate?

- High-level
  - Close to problem
  - System independent
  - Examples: Java, C#

- Low-level
  - Close to system
  - Doesn’t reflect problem
  - Examples: FORTRAN, COBOL, C++, Assembler, Machine

Compiled Languages

Human to machine readable

- Editor
- Source code file
- Compiler
- Binary executable (machine code) file

Interpreted Languages

Translation on the fly

- Editor
- Source code file
- Interpreter

How Does Java Work?

Producing portable code

- Source code is written in the Java Programming Language
- Source is compiled
  - The compiler converts Java source code into "byte code"
  - `javac prog.java` (creates `prog.class`)
- Byte code is machine code for a non-existent or virtual machine (the Java Virtual Machine – JVM)

- File
- Compiler
- Byte code
- Virtual machine code
The Two Lives of Java

Running Java Programs
- Virtual code runs on a Java virtual machine (JVM)
  - The JVM is really just an interpreter
  - An application is a local program executed on a local JVM
  - An applet is a program downloaded from the net and executed on a JVM that is a part of the web browser

```java
java prog  \( \text{reads prog.class} \)  \(<\text{applet code="prog.class"}\\text{>}
```

Hello.java : An Example

The quintessential first program

```java
public class Hello     \( // \text{class name & file name match} \\
\{
    public static void main (String[] args)
    \{
        System.out.println("Hello World");
    \}
\}
```

```javac Hello.java  \( // \text{Note that Java is case sensitive} \)
```

Program Components

Method / Function main
- Every Java and C++ program has exactly one method / function named main
  - main is not a keyword but should be treated as if it was
  - main is where the program execution begins (it is called by the system)
- Methods/functions have 4 parts
  - Name
  - Return value type
  - Argument list
  - Body

Finding and Installing the JDK

Java™ Software Development Kit
- JDK 5.0
  - Small file is an “installer” – must be connected to the internet
  - Large file is a self-extracting zip file
  - Creates a new directory named after the version (e.g., jdk1.5.0_04)
    - Recommend installing in the root (e.g., c:\jdk1.5.0_04)
  - JRE 5.0
    - Included with the JDK
  - Download separately only if you want to run but not create Java programs
    - Redistributable
    - Recommend installing in Program Files (the default)
- J2SE 5.0 Documentation
  - Install in the same directory as the JDK
  - Unzip with a program that “knows” about long file names

Completing the Java Installation

Setting environment variables
- PATH environment variable points the OS to executable files
  - A semicolon- (Windows) or colon-separated (Unix) list of directories
  - Should include Java programs (compiler, virtual machine, etc.)
- Windows XP
  - Start -> Control Panel -> Performance and Maintenance -> System
  - OR, right click My Computer -> Properties
  - Select “Advanced” tab and press “Environment Variables”
  - In System variables, select and edit “Path,” put Java at front but do not remove existing contents (press Cancel if there are any errors)
  - Press OK (if everything really is okay) to finish
- Test
  - javac -version
  - java -version