Introduction to Computing and Java
Programming Coverage

Methods, Classes, Arrays

Iteration, Control Structures

Variables, Expressions

Data Types
Course Design

• Instead
  – Lecture not a rehash of the book but covers same concepts from a different perspective
  – Lots of programming activities
Intro to Computing
FIGURE 1.9  Basic computer architecture
The CPU

• Fetches instructions from main memory
• Carries out the operations commanded by the instructions
• Each instruction produces some outcome
• A *program* is an entire sequence of instructions
• Instructions are stored as *binary numbers*
• *Binary number* - a sequence of 1’s and 0’s
Main Memory – a big list of addresses

FIGURE 1.10 Memory locations
Knowing About: Computer Hardware

• **Bit**: smallest and most basic data item in a computer; represents a 0 or a 1

• **Byte**: a grouping of eight bits
  – E.g., 00010001

• **Word**: a grouping of one or more bytes
Patterns of bits could represent integer numbers

<table>
<thead>
<tr>
<th>1 bit 4 items</th>
<th>2 bits 8 items</th>
<th>3 bits 16 items</th>
<th>4 bits 32 items</th>
<th>5 bits 64 items</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>000</td>
<td>0000</td>
<td>00000</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>001</td>
<td>0001</td>
<td>00001</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>010</td>
<td>0010</td>
<td>00100</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>011</td>
<td>0011</td>
<td>00101</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0100</td>
<td>00110</td>
<td>001010</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>0101</td>
<td>00110</td>
<td>001011</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>0110</td>
<td>00110</td>
<td>001100</td>
</tr>
<tr>
<td></td>
<td>111</td>
<td>0111</td>
<td>00110</td>
<td>001101</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1101</td>
<td>01100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1110</td>
<td>01101</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1111</td>
<td>01110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01111</td>
</tr>
<tr>
<td>Unit</td>
<td>Symbol</td>
<td>Number of Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>----------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td>$2^0 = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kilobyte</td>
<td>KB</td>
<td>$2^{10} = 1024$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>megabyte</td>
<td>MB</td>
<td>$2^{20} = 1,048,576$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gigabyte</td>
<td>GB</td>
<td>$2^{30} = 1,073,741,824$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>terabyte</td>
<td>TB</td>
<td>$2^{40} = 1,099,511,627,776$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 1.11** Units of binary storage
Bits could represent characters

**FIGURE 1.5** Text is stored by mapping each character to a number
We said that 00010001 could represent anything, a number, sound, color, etc.

**FIGURE 1.3** A sound wave and an electronic analog signal that represents the wave
FIGURE 1.4 Digitizing an analog signal by sampling
Bits can represent colors
Bits can represent instructions

- 110110
  - might be the instruction to add two numbers

- 110100
  - might be the instruction to increment a number

- Called binary code

- Assembly Code - Mnemonics

  Loop: L.D F0, 0(R1)
         ADD.D F4, F0, F2
         S.D 0(R1), F4 ; Drop DADDUI and BNEZ
         L.D F6, -8(R1)
         ADD.D F8, F6, F2
         S.D -8(R1), F8 ; Drop DADDUI and BNEZ
         L.D F10, -16(R1)
         ADD.D F12, F10, F2
The Fetch-Decode Execute Cycle

**FIGURE 1.14** The continuous fetch-decode-execute cycle
Layers of Programming Languages

<table>
<thead>
<tr>
<th>High-Level Language</th>
<th>Assembly Language</th>
<th>Machine Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>a + b</td>
<td>ld [%fp-20], %o0</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>ld [%fp-24], %o1</td>
<td>1101 0000 0000 0111</td>
</tr>
<tr>
<td></td>
<td>add %o0, %o1, %o0</td>
<td>1011 1111 1110 1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1101 0010 0000 0111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1011 1111 1110 1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001 0000 0000 0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

**FIGURE 1.21** A high-level expression and its assembly language and machine language equivalent

A program called a **compiler** translates from high-level to machine language
**Interpreter**

- Compiling combined with execution

Source Code:

```
X=3
X=X+1
...
```

Machine Language Statement:

```
11011101
```

Often easier to program, debug, but will run slower than compiled programs.
Java – Both Interpreted/Compiled

• Somewhat of a simplification with JIT compilers

```java
public class Foo {
    if (e.target == xyz) then
        this.hide();
}
```
Programming

• A program is a list of instructions for the computer to follow

• Algorithm
  – Sequence of steps to solve a problem
  – Example: Searching a list of names for a number
    1. Atto, Tom (6-1102)
    2. Attrick, Jerry (6-9089)
    3. DeBanque, Robin (6-0022)
    4. Dente, Al (6-8722)
    5. Fresco, Al (6-8723)
    6. Guini, Lynn (6-8834)
    7. Oki, Kerry (6-9213)
    8. Wright, Eaton (6-4441)
Pseudocode

• Somewhere between English and actual code to help figure out how to write the actual code

• Binary search pseudocode
  – Given a list of names
    • If the list is empty then target not found
    • Otherwise:
      – Get the name in the middle of the list
      – If this name is the same as the target, then the target is in the list
      – If this name is alphabetically before the target then
        » Repeat the process on the bottom half of the list
      – If this name is alphabetically after the target then
        » Repeat the process on the first half of the list
Java Example

• In-class: Entering and running a “Hello, World” program using Textpad

File: HelloWorld.java

/*
 * Normally you would put your name and assignment info here
 * This program prints out "Hello, World".
 */
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, world!");
    }
}