Compiling and Compiler Errors

August 22, 2007

Chapter 1
Introduction

Our First Program

```java
// comments about the class
class MyProgram {
    // comments about the method
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Java Program Structure

```java
// comments about the class
class MyProgram {
    // comments about the method
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Comments

- Comments in a program are called **inline documentation**
- They should be included to explain the purpose of the program and describe processing steps
- They do not affect how a program works
- Java comments can take three forms:
  - `//` this comment runs to the end of the line
  - `/*` this comment runs to the terminating symbol, even across line breaks `*/`
  - `/**` this is a javadoc comment `*/`
Our First Program

```java
// comments about the class
public class MyProgram {
    // comments about the method
    public static void main (String[] args) {
        System.out.println("Hello World");
    }
}
```

Identifiers

- **Identifiers** are the words a programmer uses in a program.
- An identifier can be made up of letters, digits, the underscore character (_), and the dollar sign.
- Identifiers cannot begin with a digit.
- Java is case sensitive; **Total**, **total**, and **TOTAL** are different identifiers.
- By convention, programmers use different case styles for different types of identifiers, such as:
  - **title case** for class names - **Lincoln**
  - **upper case** for constants - **MAXIMUM**

Identifiers

- Sometimes we choose identifiers ourselves when writing a program (such as **Lincoln**).
- Sometimes we are using another programmer’s code, so we use the identifiers that he or she chose (such as **println**).
- Often we use special identifiers called **reserved** words that already have a predefined meaning in the language.
- A reserved word cannot be used in any other way.

Reserved Words

- The Java reserved words:
  - **abstract**
  - **assert**
  - **boolean**
  - **break**
  - **byte**
  - **case**
  - **catch**
  - **char**
  - **class**
  - **const**
  - **continue**
  - **default**
  - **do**
  - **double**
  - **else**
  - **enum**
  - **extends**
  - **false**
  - **finally**
  - **float**
  - **for**
  - **goto**
  - **if**
  - **implements**
  - **import**
  - **instanceof**
  - **int**
  - **interface**
  - **long**
  - **native**
  - **new**
  - **null**
  - **package**
  - **private**
  - **protected**
  - **public**
  - **short**
  - **static**
  - **strictfp**
  - **super**
  - **switch**
  - **synchronized**
  - **this**
  - **throw**
  - **throws**
  - **transient**
  - **true**
  - **try**
  - **void**
  - **volatile**
  - **while**
  - **wild**
  - **yield**

White Space

- Spaces, blank lines, and tabs are called **white space**.
- White space is used to separate words and symbols in a program.
- Extra white space is ignored.
- A valid Java program can be formatted many ways.
- Programs should be formatted to enhance readability, using consistent indentation.
- See **Lincoln2.java** (page 34)
- See **Lincoln3.java** (page 35)

This code is still valid, but hard to read

```java
// comments about the class
public class MyProgram {
    // comments about the method
    public static void main (String[] args) {
        System.out.println("Hello World");
    }
}
```
Run examples from the book

Hardware and Software
- **Hardware**
  - the physical, tangible parts of a computer
  - keyboard, monitor, disks, wires, chips, etc.
- **Software**
  - programs and data
  - a program is a series of instructions
- A computer requires both hardware and software
- Each is essentially useless without the other

A Computer Specification
- Consider the following specification for a personal computer:
  - 2.8 GHz Pentium 4 Processor
  - 512 MB RAM
  - 80 GB Hard Disk
  - 48x CD-RW / DVD-ROM Combo Drive
  - 17” Video Display with 1280 x 1024 resolution
  - 56 Kb/s Modem
- What does it all mean?

CPU and Main Memory
- Chip that executes program commands
  - Intel Pentium 4
  - Sun ultraSPARC III
- Primary storage area for programs and data that are in active use
- Synonymous with RAM

The Central Processing Unit
- A CPU is on a chip called a **microprocessor**
- It continuously follows the **fetch-decode-execute cycle**:

  ![Diagram of the fetch-decode-execute cycle](Image)

  - **fetch**: Retrieve an instruction from main memory
  - **decode**: Determine what the instruction is
  - **execute**: Carry out the instruction

Secondary Memory Devices
- Information is moved between main memory and secondary memory as needed
  - Hard disks
  - Floppy disks
  - ZIP disks
  - Writable CDs
  - Writable DVDs
  - Tapes
Input / Output Devices

Central Processing Unit

Monitor
Keyboard

Main Memory

Hard Disk
Floppy Disk

Monitor screen
Keyboard
Mouse
Joystick
Bar code scanner
Touch screen

I/O devices facilitate user interaction

Software Categories

• Operating System
  • controls all machine activities
  • provides the user interface to the computer
  • manages resources such as the CPU and memory
  • Windows XP, Unix, Linux, Mac OS

• Application program
  • generic term for any other kind of software
  • word processors, missile control systems, games

• Most operating systems and application programs have a graphical user interface (GUI)

Analog vs. Digital

• There are two basic ways to store and manage data:

  • Analog
    • continuous, in direct proportion to the data represented
    • music on a record album - a needle rides on ridges in the grooves that are directly proportional to the voltages sent to the speaker

  • Digital
    • the information is broken down into pieces, and each piece is represented separately
    • music on a compact disc - the disc stores numbers representing specific voltage levels sampled at specific times

Digital Information

• Computers store all information digitally:
  • numbers
  • text
  • graphics and images
  • video
  • audio
  • program instructions

• In some way, all information is digitized - broken down into pieces and represented as numbers

Representing Text Digitally

• For example, every character is stored as a number, including spaces, digits, and punctuation

• Corresponding upper and lower case letters are separate characters

Hi, Heather.

72  105  44  32  72  101  97  116  104  101  114  46

Binary Numbers

• Once information is digitized, it is represented and stored in memory using the binary number system

• A single binary digit (0 or 1) is called a bit

• Devices that store and move information are cheaper and more reliable if they have to represent only two states

• A single bit can represent two possible states, like a light bulb that is either on (1) or off (0)

• Permutations of bits are used to store values
Bit Permutations

<table>
<thead>
<tr>
<th>1 bit</th>
<th>2 bits</th>
<th>3 bits</th>
<th>4 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>000</td>
<td>1000</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>001</td>
<td>0001</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
<td>0010</td>
<td>1010</td>
</tr>
<tr>
<td>11</td>
<td>011</td>
<td>0011</td>
<td>1011</td>
</tr>
<tr>
<td>100</td>
<td>0100</td>
<td>1100</td>
<td>1100</td>
</tr>
<tr>
<td>101</td>
<td>0101</td>
<td>1101</td>
<td>1101</td>
</tr>
<tr>
<td>110</td>
<td>0110</td>
<td>1110</td>
<td>1110</td>
</tr>
<tr>
<td>111</td>
<td>0111</td>
<td>1111</td>
<td>1111</td>
</tr>
</tbody>
</table>

Each additional bit doubles the number of possible permutations.

Each permutation can represent a particular item.

There are $2^N$ permutations of $N$ bits.

Therefore, $N$ bits are needed to represent $2^N$ unique items.

How many items can be represented by:
- 1 bit: $2^1 = 2$ items
- 2 bits: $2^2 = 4$ items
- 3 bits: $2^3 = 8$ items
- 4 bits: $2^4 = 16$ items
- 5 bits: $2^5 = 32$ items

More about binary numbers later...

Program Development

- The mechanics of developing a program include several activities:
  - writing the program in a specific programming language (such as Java)
  - translating the program into a form that the computer can execute
  - investigating and fixing various types of errors that can occur
  - Software tools can be used to help with all parts of this process

Programming Languages

- Each type of CPU executes only a particular machine language
- A program must be translated into machine language before it can be executed
- A **compiler** is a software tool which translates source code into a specific target language
- Often, that target language is the machine language for a particular CPU type
- The Java approach is somewhat different

Java Translation

- The Java compiler translates Java source code into a special representation called bytecode
- Java bytecode is not the machine language for any traditional CPU
- Another software tool, called an **interpreter**, translates bytecode into machine language and executes it
- Therefore the Java compiler is not tied to any particular machine
- Java is considered to be architecture-neutral
Java Translation

Syntax and Semantics

• The syntax rules of a language define how we can put together symbols, reserved words, and identifiers to make a valid program.

• The semantics of a program statement define what that statement means (its purpose or role in a program).

• A program that is syntactically correct is not necessarily logically (semantically) correct.

• A program will always do what we tell it to do, not what we meant to tell it to do.

Basic Program Development

Errors

• A program can have three types of errors:

  • The compiler will find syntax errors and other basic problems (compile-time errors).
  
    ▪ If compile-time errors exist, an executable version of the program is not created.

  • A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally (run-time errors).

  • A program may run, but produce incorrect results, perhaps using an incorrect formula (logical errors).

Development Environments

• There are many programs that support the development of Java software, including:
  
    ▪ Sun Java Development Kit (JDK)
    ▪ Sun NetBeans
    ▪ IBM Eclipse
    ▪ Borland JBuilder
    ▪ MetroWerks CodeWarrior
    ▪ BlueJ
    ▪ jGRASP

  • Though the details of these environments differ, the basic compilation and execution process is essentially the same.

HW 1 is out

• Posted on the class web page.
THE END