Chapter Goals

- To learn about computers and programming
- To compile and run your first Java program
- To recognize compile-time and run-time errors
- To describe an algorithm with pseudocode

In this chapter, you will learn how to write and run your first Java program. You will also learn how to diagnose and fix programming errors, and how to use pseudocode to describe an algorithm.

Contents

- Computer Programs
- The Anatomy of a Computer
- The Java Programming Language
- Becoming Familiar with your Programming Environment
- Analyzing Your First Program
- Errors
- Problem Solving:
  - Algorithm Design

1.1 Computer Programs

- A Computer Program is a sequence of instructions and decisions
- Computers execute very basic instructions in rapid succession
- Programming is the act of designing and implementing computer programs
1.2 The Anatomy of a Computer

- The central processing unit (CPU) performs program control and data processing.
- Storage devices include memory (RAM) and secondary storage:
  - Hard disk
  - Flash drives
  - CD/DVD drives
- Input/Output devices allow the user to interact with the computer:
  - Mouse, keyboard, printer, screen…

When you ‘run’ a program

- Program instructions and data (such as text, numbers, audio, or video) are stored on the hard disk, a compact disk (or DVD), or elsewhere on the network.
- When a program is started, it is brought into memory, where the CPU can read it.
- The CPU runs the program one instruction at a time. The program may react to user input.
- As directed by these instructions and the user, the CPU reads data, modifies it, and writes it back to memory, the screen or secondary storage.

1.3 The Java Language

- In 1991, James Gosling of Sun Microsystems designed what would become the Java programming language.
- Java was originally designed for programming consumer devices, but it was first successfully used to write Internet applets:
  - An applet is typically embedded inside a web page and runs in the context of a browser.
Java History

- Java Design Goals
  - Safe: Can be run inside a browser and will not attack your computer
  - Portable: Run on many Operating Systems
    - Linux, Windows, Mac OS, Android
- Java programs are distributed as instructions for a 'virtual machine,' making them platform-independent
  - Virtual machines are available for most Operating Systems. The iPhone is a notable exception

Java Virtual Machines

- Source code
  - The compiler generates byte code in a "class" file which can be run on any Java Virtual Machine
- Portable 'byte code'
  - The compiler generates byte code in a "class" file which can be run on any Java Virtual Machine

Java Timeline

<table>
<thead>
<tr>
<th>Version</th>
<th>Year</th>
<th>Important New Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1996</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>1997</td>
<td>Inner classes</td>
</tr>
<tr>
<td>1.2</td>
<td>1998</td>
<td>Swing, Collections framework</td>
</tr>
<tr>
<td>1.3</td>
<td>2000</td>
<td>Performance enhancements</td>
</tr>
<tr>
<td>1.4</td>
<td>2002</td>
<td>Assertions, XML support</td>
</tr>
<tr>
<td>5</td>
<td>2004</td>
<td>Generic classes, enhanced for loop, auto-boxing, enumeration, annotations</td>
</tr>
<tr>
<td>6</td>
<td>2006</td>
<td>Library improvements</td>
</tr>
<tr>
<td>7</td>
<td>2011</td>
<td>Small language changes and library improvements</td>
</tr>
</tbody>
</table>

- Oracle purchased Sun (along with Java) in 2010
  - There are still quite a few references and links to Sun Microsystems which are now re-directed to Oracle

The Java API

- The Java Platform consists of two parts:
  1) Java Virtual Machine
  2) Java API
    -- also called libraries
- The Application Programming Interface (API) is a huge collection of handy software packages that programmers can use:
  - Graphics, user interface, networking, sound, database, math, and many more
Your First Program

- Traditional ‘Hello World’ program in Java

```java
public class HelloPrinter {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

- We will examine this program in the next section
  - Java is Case Sensitive
  - Java uses special characters, e.g. `{ } ( ) ;

Text Editor Programming

- You can also use a simple text editor such as Notepad to write your source code
- Once saved as HelloPrinter.java, you can use a console window to:
  1. Compile the program
  2. Run the program

Source Code to Running Program

- The compiler generates the .class file which contains instructions for the Java Virtual machine
- Class files contain ‘byte code’ that you cannot edit
  - `D:\temp\hello \Type HelloPrinter.class`
  - `\D:\temp\hello \Type HelloPrinter.class`
  - `Hello, World! HelloPrinter.java`
1.5 Analyzing your First Program

1. public class HelloPrinter
2. {
3.   public static void main(String[] args)
4.   {
5.     System.out.println("Hello, World!");
6.   }
7. }

1: Declares a 'class' HelloPrinter
   -- Every Java program has one or more classes.
3: Declares a method called 'main'
   -- Every Java application has exactly one ‘main’ method
   -- Entry point where the program starts
5: Method System.out.println outputs ‘Hello, World!’
   -- A statement must end with a semicolon (;)

Syntax 1.1: The Java Program

- Every application has the same basic layout
  - Add your ‘code’ inside the main method

Calling Java Library methods

- Line 5 shows how to ‘call’ a ‘method’ from the Java API: System.out.println
  - Code that somebody else wrote for you!
  - Notice the dots (periods)
  - Parenthesis surround the arguments that you ‘pass’ to a method ("Hello, World!");
  - We are passing a String “Hello World”
    - Note the double quotes which denote a String inside
  - You can also print numerical values
    - System.out.println(3 + 4);

Getting to know println

- The println method prints a string or a number and then starts a new line.
  System.out.println("Hello");
  System.out.println("World!");

- println has a ‘cousin’ method named print that does not print a new line.
  System.out.print("00");
  System.out.println(3+4);
Common Error 1.1

- **Omitting Semicolons**
  - In Java, every statement must end in a semicolon. Forgetting to type a semicolon is a common error. It confuses the compiler, because the compiler uses the semicolon to find where one statement ends and the next one starts. For example, the compiler sees this:
    ```java
    System.out.println("Hello")
    System.out.println("World!");
    ```
  - As this:
    ```java
    System.out.println("Hello") System.out.println("World!");
    ```
  - It doesn’t understand this statement, because it does not expect the word `System` following the closing parenthesis after `Hello`.

1.6 Errors

- **The Two Categories of Errors:**
  1) **Compile-time Errors**
     - **Syntax Errors**
       - Spelling, Capitalization, punctuation
       - Ordering of statements, matching of braces/parenthesis...
     - No `.class` file is generated by the compiler
     - Correct first error listed, then compile again
  2) **Run-time Errors**
     - Logic Errors, `.class` file is generated by the compiler
     - Program runs, but produces unintended results
     - Program may ‘crash’

Syntax Errors

- What happens if you
  - Misspell a word: `System.out.println`
  - Don’t Capitalize a word: `System.out.println`
  - Leave out a word: `void`
  - Forget a Semicolon after: `System.out.println("Hello, World!");` ("Hello, World!")
  - Don’t match a curly brace? Remove line 6
- Try it to see what error messages are generated

Logic Errors

- What happens if you
  - Divide by Zero: `System.out.println(1/0);`
  - Mis-spell output: `System.out.println("Hello, Word!");` ("Hello, World!")
  - Forget to output: Remove line 5
- Programs will compile and run
  - The output may not be as expected
1.7 Problem Solving: Algorithm Design

- Algorithms are simply plans
  - Detailed plans that describe the steps to solve a specific problem
- You already know quite a few
  - Calculate the area of a circle
  - Find the length of the hypotenuse of a triangle
- Some problems are more complex and require more steps
  - Calculate PI to 100 decimal places
  - Calculate the trajectory of a missile

Text Problem to Algorithm

- Given the problem:
  - You put $10,000 into a bank account that earns 5 percent interest per year. How many years does it take for the account balance to be double the original?

- How would you solve it?
  - Manual method
    - Make a table
    - Add lines until done
  - Use a spreadsheet!
    - Write a formula
      - Per line, based on line above

Text Problem to Algorithm Steps

- You put $10,000 into a bank account that earns 5 percent interest per year. How many years does it take for the account balance to be double the original?

- Break it into steps
  - Start with a year value of 0 and a balance of $10,000
  - Repeat the following while the balance is less than $20,000
  - Add 1 to the year value
  - Multiply the balance by 1.05 (5% increase)

Text Problem to Pseudocode

- Pseudocode
  - Half-way between natural language and a programming language

- Modified Steps
  - Set the year value of 0
  - Set the balance to $10,000
  - While the balance is less than $20,000
    - Add 1 to the year value
    - Multiply the balance by 1.05
  - Report the final year value as the answer

- This can be translated into Java fairly easily
Algorithm Defined

- An algorithm describes a sequence of steps that is:
  - Unambiguous
    - Do not require ‘assumptions’
    - Uses precise instructions
  - Executable
    - Can be carried out in practice
  - Terminating
    - Will eventually come to an end

Steps: Algorithm to Pseudocode

1. Determine the inputs and outputs
   From the problem statement:
   \[ \text{price1, price2, mpg1, mpg2} \]

2. Break down the problem into smaller tasks
   ‘Calculate total cost’ for each car

3. Describe each subtask as pseudocode
   \[ \text{total cost} = \text{purchase price} + \text{operating cost} \]

4. Test your pseudocode with example input

Summary: Computer Basics

- Computers execute very basic instructions in rapid succession.
- A computer program is a sequence of instructions and decisions.
- Programming is the activity of designing and implementing computer programs to perform a certain task.
- The central processing unit (CPU) performs program control and data processing.
- Storage devices include memory and secondary storage.

Summary: Java

- Java was originally designed for programming consumer devices, but it was first successfully used to write Internet applets.
- Java was designed to be safe and portable.
- Java programs are distributed as instructions for a virtual machine, making them platform-independent.
- Java has a very large set of libraries. Focus on learning those parts of libraries that you need for your programming applications.
Summary: Java

- Classes are the fundamental building blocks of Java programs.
- Every Java application contains a class with a main method. When the application starts, the instructions in the main method are executed.
- Each class contains declarations of methods. Each method contains a sequence of instructions.
- A method is called by specifying the method and its parameters.
- A string is a sequence of characters enclosed in quotation marks.

Summary: Errors and Pseudocode

- A compile-time error is a violation of the programming language rules that is detected by the compiler.
- A run-time error causes a program to take an action that the programmer did not intend.
- Pseudocode is an informal description of a sequence of steps for solving a problem.
- An algorithm for solving a problem is a sequence of steps that is unambiguous, executable, and terminating.