Chapter 1
Introduction to Computers, Programs, and Java

1.1 Introduction

- Java is the **Internet** program language
- Why Java? The answer is that Java enables users to deploy applications on the Internet for **servers**, **desktop computers**, and **small hand-held devices**.

1.2 What is a Computer?

- A computer is an electronic device that stores and processes data.
- A computer includes both hardware and software.
  - **Hardware** is the physical aspect of the computer that can be seen.
  - **Software** is the invisible instructions that control the hardware and make it work.
- **Computer programming** consists of writing instructions for computers to perform.
- A computer consists of the following hardware components
  - CPU (Central Processing Unit)
  - Memory (Main memory)
  - Storage Devices (hard disk, floppy disk, CDs)
  - Input/Output devices (monitor, printer, keyboard, mouse)
  - Communication devices (Modem, NIC (Network Interface Card)).
1.2.1 Central Processing Unit (CPU)

- The central processing unit (CPU) is the **brain** of a computer.
- It retrieves instructions from memory and executes them.
- The CPU usually has two components: a **control Unit** and **Arithmetic/Logic Unit**.
- The control unit coordinates the actions of the other components.
- The ALU (Arithmetic/Logic Unit) unit performs numeric operations (+, -, /, *) and logical operations (comparison).
- The CPU speed is measured by clock speed in megahertz (MHz), with 1 megahertz equaling 1 million pulses per second.
- The speed of the CPU has been improved continuously.
- If you buy a PC now, you can get an Intel Pentium 4 Processor at 3 gigahertz (1 gigahertz is 1000 megahertz).
1.2.2 Memory

- Computers use zeros and ones because digital devices have two stable states (on / off)
- Data of various kinds, such as numbers, characters and strings, are encoded as series of bits (binary digits: zeros and ones).

<table>
<thead>
<tr>
<th>Memory address</th>
<th>Memory content</th>
<th>Encoding for character 'J'</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>01001010</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>01100001</td>
<td>Encoding for character 'a'</td>
</tr>
<tr>
<td>2002</td>
<td>01110110</td>
<td>Encoding for character 'v'</td>
</tr>
<tr>
<td>2003</td>
<td>01100001</td>
<td>Encoding for character 'a'</td>
</tr>
<tr>
<td>2004</td>
<td>00000011</td>
<td>Encoding for number 3</td>
</tr>
</tbody>
</table>

FIGURE 1.2 Memory stores data and program instructions

- Memory is to store data and program instructions for CPU to execute.
- A memory unit is an ordered sequence of bytes, each holds eight bits.
- A programmer need not be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme.
- The encoding scheme varies; for example, ‘J’ is represented by 01001010 in one byte by ASCII encoding.
- If a computer needs to store a large number that cannot fit into a single byte, it uses several adjacent bytes. No two data items can be share or split the same byte.
- A byte is the minimum storage unit.
- A program and its data must be brought to memory before they can be executed.
- A memory byte is never empty, but its initial content may be meaningless to your program.
- The current content of a memory byte is lost whenever new information is placed in it.
- Every byte has a unique address. The address is used to locate the byte for storing and retrieving data.
- Since bytes can be accessed at any location, the memory is also referred to as RAM (random-access memory).
- Memory chips are slower and less expensive than CPU chips.
- Computer storage size is measured in bytes, kilobytes (KB), megabytes (MB), gigabytes (GB), and terabytes (TB).
  - A kilobyte is $2^{10} = 1024$, about 1000 bytes
  - A terabyte is about 1000 gigabytes
1.2.3 Storage Devices

- Memory is volatile, because information is lost when the power is off.
- Programs and data are permanently stored on storage devices and are moved to memory when the computer actually uses them. The reason for that is that memory is much faster than storage devices.
- There are four main types of storage devices:
  - Disk drives (hard disks and floppy disks)
  - CD drives (CD-R, CD-RW, and DVD)
  - Tape drives
  - USB flash drives

1.2.4 Input and Output Devices

- The common input devices are keyboard and mouse.
- The common output devices are printers and monitors.

1.2.5 Communication Devices

- Commonly used communication devices are:
  - **Dialup modem**: A dialup modem uses a phone line and can transfer data at a speed up to 56,000 bps (bits per second).
  - **DSL**: A DSL (digital subscriber line) also uses a phone line and can transfer data at a speed 20 times faster than a dialup modem.
  - **Cable modem**: A cable modem uses the TV cable line maintained by the cable company. A cable modem is as fast as DSL.
  - **Network Interface Card (NIC)**: A network interface card (NIC) is a device that connects a computer to a local area network (LAN). A typical NIC called 10BaseT can transfer data at 10 mbps (million bits per second).
1.3 Programs

- Computer programs, known as **software**, are instructions to the computer.
- You tell a computer what to do through programs. Without programs, a computer is an empty machine.
- Computers do not understand human languages, so you need to use computer languages to communicate with them.
- Programs are written using **programming languages**.
- The language a computer speaks is machine language.
- Machine Language is a set of primitive instructions built into every computer. Machine languages are **different** for **different** type of computers.
- The instructions are in the form of **binary** code, so you have to enter binary codes for various instructions.
- Program with native machine language is a tedious process. Moreover, the programs are highly difficult to read and modify.
- For example, to add two numbers, you might write an instruction in binary like this:

  \[1101101010011010\]

- Assembly Language is a **low-level** language in which a mnemonic is used to represent each of the machine language instructions.
- Assembly languages were developed to make programming easy.
- Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code.
- For example, to add two numbers, you might write an instruction in assembly code like this:

**ADDF3 R1, R2, R3**

![Assembly Source File](image)

![Assembler](image)

![Machine Code File](image)

**FIGURE 1.3** Assembler translates assembly language instructions to machine code.
• The **high-level languages** are **English-like** and easy to learn and program.
• They were developed to overcome the platform-specific problem.
• For example, the following is a high-level language statement that computes the area of a circle with radius 5:

\[
\text{area} = 5 * 5 * 3.1415;
\]

• There are more than one hundred languages; the most popular of them are:
  o COBOL (CCommon Business Oriented Language)
  o FORTRAN (FORmula TRANslation)
  o BASIC (Beginner All-purpose Symbolic Instructional Code)
  o Pascal (named for Blaise Pascal)
  o Ada (named for Ada Lovelace)
  o C (whose developer designed B first)
  o Visual Basic (Basic-like visual language developed by Microsoft)
  o Delphi (Pascal-like visual language developed by Borland)
  o C++ (an object-oriented language, based on C)
  o C# (a Java-like language developed by Microsoft)
  o Java (We use it in the book)

• Each language was designed with a specific purpose:
  o COBOL was designed for **business** applications.
  o FORTRAN for **mathematical** computations.
  o Pascal to be a simple structural programming language.
  o BASIC was designed to be learned and used **easily**.
  o C combines the power of an assembly language with the ease of use and portability of a high-level language.
  o Visual Basic and Delphi are used in developing **graphical user interfaces** and in rapid application development.
  o C++ is popular for **system software** projects like writing compiler and operating systems. The Microsoft Windows operating system was coded using **C++**.
• A program written in a high-level language is called a **source** program.
• Since a computer cannot understand a source program, a program called a **compiler** is used to **translate** the source program into a machine-language program.
• The machine-language program is often then linked with other supporting **library code** to form an executable file.
• The executable file can be executed on the machine. On windows, executable files have extension `.exe`.

![Diagram of compilation process](image)

**FIGURE 1.4** A source program is compiled into a machine-language file, which is then linked with the system library to form an executable file

• You can port a source program to any machine with appropriate compilers.
• The source program must be recompiled, however, because the object program can **only** run on a specific machine.
• Java was designed to run object programs on **any** platform.
• With Java, you write the program once, and compile the source program into a special type of object code, known as **bytecode**.
• The bytecode can then run on any computer with a **Java Virtual Machine (JVM)**, as shown in figure below. Java Virtual Machine is software that **interprets** Java bytecode.

![Diagram of Java execution process](image)

**FIGURE 1.11** Java byte code can be executed on any computer with a Java Virtual Machine.
### 1.4 Operating Systems

- The Operating System (OS) is **the most important program** that manages and controls a computer’s activities. You are probably using Microsoft Windows, Mac OS, and Linux.
- Application programs such as an Internet browser and a word processor cannot run without an operating system.

![Diagram of the operating system](image)

**FIGURE 1.5** The operating system is the software that controls and manages the system.

- The major tasks of the OS are:
  - Controlling and monitoring system activities
  - Allocating and assigning system resources
  - Scheduling operations (Multiprogramming, Multithreading, Multiprocessing)
    - Multiprogramming allows **multiple** programs to run simultaneously by sharing the CPU. For example, you may use a word processor to edit a file while the Web browser is downloading a file at the same time.
    - Multithreading allows concurrency within **a program**, so that its subunits can run at the same time. For example, editing and saving are two tasks with the same application.
    - Multiprocessing, or parallel processing uses **two or more processors** together to perform **a task**. It is like a surgical operation where several doctors work together on one patient.
1.5 Java, World Wide Web, and Beyond

- Developed by a team led by James Gosling at Sun Microsystems. Originally called oak (1991) for use in embedded consumer electronic applications.
- In 1995, renamed Java, it was redesigned for developing Internet applications.
- Java is a general purpose programming language.
- Java is the Internet programming language.
- Today, Java is used in not only for Web programming, but also developing standalone applications across platforms on servers, desktop computers, and mobile devices.
- Java programs can be embedded in HTML pages and downloaded by Web browsers to bring live animation and interaction to web clients.
- Java initially became attractive because Java programs can be run from a Web browser. Java programs that run from a Web browser are called applets.

![A Java Applet for playing TicTacToe is embedded in an HTML page.](image-url)
• Java can also be used to develop applications on the server side. These applications can be run from a Web server to generate dynamic Web pages.

![Java LiveLab BE](image)

FIGURE 1.7 Java was used to develop an automatic grading system to accompany this book.

• Java can also be used to develop applications for hand-held devices such as Palm and cell phones

![Hand-held Devices](image)

FIGURE 1.8 Java can be used to develop applications for hand-held and wireless devices, such as a PDA (left) and a cell phone (right).
1.6 The Java Language Specification, API, JDK, and IDE

- **API**: The Application Program Interface (API) contains predefined classes and interfaces for developing Java programs.
- **JDK Versions**: Sun releases each version with of a Java Development Toolkit (JDK).
  - JDK 1.02 (1995)
  - JDK 1.1 (1996)
  - JDK 1.2 (1998)
  - JDK 1.3 (2000)
  - JDK 1.4 (2002)
  - JDK 1.5 (2004) = JDK 5 or Java 5
  - JDK 1.6 (2006) = JDK 6 or Java 6
  - JDK 1.7 (possibly 2010) = JDK 7 or Java 7
- **Java Development Toolkit (JDK) Editions**
  - Java Standard Edition (Java SE): Java SE can be used to develop client-side standalone applications or applets.
  - Java Enterprise Edition (Java EE): Java EE can be used to develop server-side applications such as Java servlets and Java ServerPages.
  - Java Micro Edition (Java ME): Java ME can be used to develop applications for mobile devices such as cell phones.
- **This book uses **Java SE 6** to introduce Java programming.**
- **JDK consists of a set of separate programs for developing and testing Java programs, each of which is invoked from a command line.**
- **There are tools that provide an Integrated Development Environment (IDE) for rapidly developing Java programs. Editing, compiling, building, debugging, and online help are integrated in one GUI.**
  - JBuilder by Borland ([www.borland.com](http://www.borland.com))
  - NetBeans Open Source by Sun ([www.netbeans.org](http://www.netbeans.org))
  - Eclipse, Open Source by IBM ([www.eclipse.org](http://www.eclipse.org))
  - Code Warrior by Metrowerks ([www.metrowerks.com](http://www.metrowerks.com))
  - TextPad Editor ([www.textpad.com](http://www.textpad.com))
  - JCreator LE ([www.jcreator.com](http://www.jcreator.com))
  - JEdit ([www.jedit.org](http://www.jedit.org))
  - JGrasp ([www.jgrasp.org](http://www.jgrasp.org))
  - BlueJ ([www.bluej.org](http://www.bluej.org))
  - DrJava ([http://drjava.sourceforge.net](http://drjava.sourceforge.net))
1.7 A Simple Java Program

    //This application program prints Welcome to Java!
    public class Welcome
    {
        public static void main(String[] args)
        {
            System.out.println("Welcome to Java!");
        }
    }
1.8 Creating, Compiling, and Executing a Java Program

- You have to create your program and compile it before it can be executed. You can use any text editor or IDE to create and edit a Java source-code. This process is iterative. Ex. Save your file as “Welcome.java”
- If your program has compilation errors, you have to fix them by modifying the program, and then recompile it. Ex. `javac Welcome.java`
- If your program has runtime errors or does not produce the correct results, you have to modify the programs, recompile it, and execute it again. Ex. `java Welcome`

![Diagram of Java programming-development process]

FIGURE 1.9 The Java programming-development process consists of creating/modifying source code, compiling, and executing programs.
• Java was designed to run object programs on any platform.
• With Java, you write the program once, and compile the source program into a special type of object code, known as bytecode.
• The bytecode can then run on any computer with a Java Virtual Machine (JVM), as shown in figure below. Java Virtual Machine is software that interprets Java bytecode.

![Diagram of Java bytecode and Java Virtual Machine]

FIGURE 1.11 Java byte code can be executed on any computer with a Java Virtual Machine.
Anatomy of a Java Program

• Comments
  o In Java, comments are preceded by two slashes (//) in a line, or enclosed between /* and */ in one or multiple lines. When the compiler sees //, it ignores all text after // in the same line. When it sees /*, it scans for the next */ and ignores any text between /* and */.
  Example: /* This is just a comment */

• Reserved Words
  o Reserved words or keywords are words that have a specific meaning to the compiler and cannot be used for other purposes in the program.
  o For example, when the compiler sees the word class, it understands that the word after class is the name for the class. Other reserved words in the previous example are public, static, and void. Their use will be introduced later.

• Modifiers
  o Java uses certain reserved words called modifiers that specify the properties of the data, methods, and classes and how they can be used.
  o Examples of modifiers are public and static. Other modifiers are private, final, abstract, and protected. A public datum, method, or class can be accessed by other classes. A private datum or method cannot be accessed by other classes.

• Statements
  o A statement represents an action or a sequence of actions.
  o The statement System.out.println("Welcome to Java!") in the program is a statement to display the greeting "Welcome to Java!" Every statement in Java ends with a semicolon (;

• Blocks
  o A pair of braces in a program forms a block that groups components of a program.
  o In Java, each block begins with an open brace ({ ) and ends with a closing brace (}).
  o Every class has a class block that groups the data and methods of the class.
  o Every class has a method block that groups the statements in the method.
  o Blocks can be nested, meaning that one block can be placed within another.

• Classes
  o The class is the essential Java construct. A class is a template or blueprint for objects.
  o To program in Java, you must understand classes and be able to write and use them. The mystery of the class will continue to be unveiled throughout this book. For now, though, understand that a program is defined by using one or more classes.
  o The program file must end with the extension .java and should have the exact same name as the public class name. So the file name must be Welcome.java. Every Java program has at least one class.

• Methods
  o What is System.out.println? System.out is known as the standard output object. println is a method: a collection of statements that performs a sequence of operations to display a message on the console. It can be used even without fully understanding the details of how it works. It is used by invoking a statement with a string argument. The string argument is enclosed within parentheses. In this case, the argument is "Welcome to
Java!" You can call the same println method with a different argument to print a different message.

- The main Method
  - The main method provides the control of program flow. The Java interpreter executes the application by invoking the main method.
  - Every Java application must have a user-declared main method that defines where the program begins.
  - The main method looks like this:
    
    ```java
    public static void main(String[] args) {
      // Statements;
    }
    ```

    // Welcome.java: This application program prints Welcome to Java!

    ```java
    public class Welcome {
      public static void main(String[] args) {
        System.out.println("Welcome to Java!");
      }
    }
    ```
1.9 (GUI) Displaying Text in a Message Dialog Box

- You can use the `showMessageDialog` method in the `JOptionPane` class. `JOptionPane` is one of the many predefined classes in the Java system, which can be reused rather than “reinventing the wheel.”

**NOTE:** When your program starts, a thread is spawned to run the program. When the `showMessageDialog` is invoked, a separate thread is spawned to run this method. The thread is not terminated even you close the dialog box. To terminate the thread, you have to invoke the exit method.

```java
/* WelcomeInMessageDialogBox.java: 
This application program displays Welcome to Java! 
in a message dialog box. */

import javax.swing.JOptionPane;

public class WelcomeInMessageDialogBox {
    public static void main(String[] args) {
        // Display Welcome to Java! in a message dialog box
        JOptionPane.showMessageDialog(null, "Welcome to Java!", "Example 1.2 Output", JOptionPane.INFORMATION_MESSAGE);

        // Exit the program
        System.exit(0);
    }
}
```

**The showMessageDialog Method**

`JOptionPane.showMessageDialog`

- `null`, // center of screen
- "Welcome to Java!", // Message
- "Example 1.2 Output", // Title
- JOptionPane.INFORMATION_MESSAGE // icon

**The Exit Method**

- Use Exit to terminate the program and stop all threads.
Two Ways to Invoke showMessageDialog Method

- There are several ways to use the showMessageDialog method. For the time being, all you need to know are two ways to invoke it.
- One is to use a statement as shown in the example:

  ```java
  JOptionPane.showMessageDialog(null, x, y, JOptionPane.INFORMATION_MESSAGE);
  ```

  where `x` is a string for the text to be displayed, and `y` is a string for the title of the message dialog box.
- The other is to use a statement like this:

  ```java
  JOptionPane.showMessageDialog(null, x);
  ```

  where `x` is a string for the text to be displayed.