Chapter 1
Introduction to Computers, Programs, and Java

1.1 Introduction

- The central theme of this book is to learn how to solve problems by writing a program.
- This book teaches you how to create programs by using the Java programming languages.
- Java is the Internet program language
- Why Java? The answer is that Java enables user 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)).

![Diagram of computer components]

FIGURE 1.1 A computer consists of a CPU, memory, Hard disk, floppy disk, monitor, printer, and communication devices.
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.
- Intel's newest processors run at 3 about gigahertz (1 gigahertz is 1000 megahertz).

1.2.2 Bits and Bytes

- A computer is really nothing more than a series of switches.
- Each switch exists in two states: on or off.
- If the switch is on, its value is 1. If switch is off, its value is 0. These 0s and 1s are interpreted as digits in the binary number system and are called bits (binary digits).
- The minimum storage unit in a computer is a byte. A byte is composed of eight bits.
- 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 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>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>01001010</td>
</tr>
<tr>
<td>2001</td>
<td>01100001</td>
</tr>
<tr>
<td>2002</td>
<td>01110110</td>
</tr>
<tr>
<td>2003</td>
<td>01100001</td>
</tr>
<tr>
<td>2004</td>
<td>00000011</td>
</tr>
</tbody>
</table>

Encoding for character ‘J’
Encoding for character ‘a’
Encoding for character ‘v’
Encoding for character ‘a’
Encoding for number 3

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.
1.2.4 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)
  - Tape drives
  - USB flash drives

1.2.5 Input and Output Devices

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

1.2.6 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 Programming Languages

- 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.

1.3.1 Machine 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:

  1101010101011010

1.3.2 Assembly Language

- 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:

  ADD 2, 3, result

![FIGURE 1.3 An assembler translates assembly language instructions to machine code.](image-url)
1.3.3 High-Level Language

- 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:

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

- There are more than one hundred languages; the most popular of them are:

<table>
<thead>
<tr>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada</td>
<td>Named for Ada Lovelace, who worked on mechanical general-purpose computers. The Ada language was developed for the Department of Defense and is used mainly in defense projects.</td>
</tr>
<tr>
<td>BASIC</td>
<td>Beginner’s All-purpose Symbolic Instruction Code. It was designed to be learned and used easily by beginners.</td>
</tr>
<tr>
<td>C</td>
<td>Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.</td>
</tr>
<tr>
<td>C++</td>
<td>C++ is an object-oriented language, based on C.</td>
</tr>
<tr>
<td>C#</td>
<td>Pronounced “C Sharp.” It is a hybrid of Java and C++ and was developed by Microsoft.</td>
</tr>
<tr>
<td>COBOL</td>
<td>COmmon Business Oriented Language. Used for business applications.</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>FORmula TRANslation. Popular for scientific and mathematical applications.</td>
</tr>
<tr>
<td>Java</td>
<td>Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.</td>
</tr>
<tr>
<td>Pascal</td>
<td>Named for Blaise Pascal, who pioneered calculating machines in the seventeenth century. It is a simple, structured, general-purpose language primarily for teaching programming.</td>
</tr>
<tr>
<td>Python</td>
<td>A simple general-purpose scripting language good for writing short programs.</td>
</tr>
<tr>
<td>Visual Basic</td>
<td>Visual Basic was developed by Microsoft and it enables the programmers to rapidly develop graphical user interfaces.</td>
</tr>
</tbody>
</table>
• A program written in a high-level language is called a **source** program.
• Because a computer cannot execute a source program, a source program must be **translated** into machine code for execution.
• The translation can be done using another programming tool called an interpreter or a compiler:
  o An **interpreter** reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in the figure 1.4 (a). Note that a statement from the source code may be translated into several machine instructions.
  o A **compiler** translates the entire source code into a machine-code file, and the machine-code file is then executed, as shown in the figure 1.4b.

**FIGURE 1.4** (a) An interpreter translates and executes a program one statement at a time. (b) A compiler translates the entire program into a machine-language file for execution.
1.4 Operating Systems

- The Operating System (OS) is the most important program that manages and controls a computer’s activities. The popular operating systems for general-purpose computers are 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 User, Application Programs, Operating System, and Hardware]

FIGURE 1.5 Users and applications access the computer's hardware via the operating 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.
- Early History Website: http://java.sun.com/features/1998/05/birthday.html
- 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 can be used to develop applications on the server side. These applications can be run from a Web server to generate dynamic Web pages.
- Java can be used to develop applications for hand-held devices such as Palm and cell phones.
- 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.
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 (2011) = JDK 7 or Java 7
  - JDK 1.8 (2014) = JDK 8 or Java 8
- Java Development Toolkit (JDK) Editions
  - Java Standard Edition (Java SE): It can be used to develop client-side standalone applications or applets.
  - Java Enterprise Edition (Java EE): It can be used to develop server-side applications such as Java servlets, Java ServerPages (JSP), and Java ServerFaces (JSF).
  - Java Micro Edition (Java ME): It can be used to develop applications for mobile devices such as cell phones.
- This book uses Java SE 8 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)
  - NetBeans Open Source by Sun (www.netbeans.org)
  - Eclipse, Open Source by IBM (www.eclipse.org)
  - Code Warrior by Metrowerks (www.metrowerks.com)
  - TextPad Editor (www.textpad.com)
  - JCreator LE (www.jcreator.com)
  - JEdit (www.jedit.org)
  - JGrasp (www.jgrasp.org)
  - BlueJ (www.bluej.org)
  - DrJava (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!");
    }
}

<table>
<thead>
<tr>
<th>Character Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{}</td>
<td>Opening and closing braces Denotes a block to enclose statements.</td>
</tr>
<tr>
<td>()</td>
<td>Opening and closing parentheses Used with methods.</td>
</tr>
<tr>
<td>[]</td>
<td>Opening and closing brackets Denotes an array.</td>
</tr>
<tr>
<td>//</td>
<td>Double slashes Precedes a comment line.</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>Opening and closing quotation marks Enclosing a string (i.e., sequence of characters).</td>
</tr>
<tr>
<td>;</td>
<td>Semicolon Marks the end of a statement.</td>
</tr>
</tbody>
</table>
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`

FIGURE 1.6 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.

FIGURE 1.8 (a) Java byte code is translated into bytecode. (b) Java bytecode can be executed on any computer with a Java Virtual Machine.
1.9 Programming Style and Documentation

- Programming Style deals with what programs look like.
- Documentation is the body of explanatory remarks and comments pertaining to a program.
- Programming style and documentation are as important as coding. They make the programs easy to read.

1.9.1 Appropriate Comments and Comments Style

- Include a summary at the beginning of the program to explain what the program does, its key features, its supporting data structures, and unique techniques it uses.
- In a long program, you should also include comments that introduce each major step and explain anything that is difficult to read.
- Make your comments concise so they do not crowd the program or make it difficult to read.
- Include your name, class section, date, instruction, and a brief description at the beginning of the program.

1.9.2 Proper Indentation and Spacing

- **Indentation** is used to illustrate the structural relationships between program’s components or statements.
- Indent two spaces in each subcomponent more than the structure which it is nested.
- Use a single space on both sides of a binary operator.

  boolean b = 3 + 4 * 4 > 5 * (4 + 3)

- Use a blank line to separate segments of the code.

1.9.3 Block Styles

- A block is a group of statements surrounded by braces. Use end-of-line style for braces or next-line style.

```java
public class Test {
    public static void main(String[] args) {
        System.out.println("Block Styles");
    }
}
```
1.10 Programming Errors

1.10.1 Syntax Errors

- Errors that occur during compilation are called syntax errors or compilation errors.
- Syntax errors result from errors in code construction, such as mistyping a keyword, omitting necessary punctuation, or using an opening brace without a corresponding closing brace.
- These errors are easily detected, because the compiler tells you where they are and the reasons for them.

```java
public class ShowSyntaxErrors {
    public static void main(String[] args) {
        i = 30;
        System.out.println(i+4);
    }
}
```

1.10.2 Runtime Errors

- Runtime errors are errors that cause a program to terminate abnormally.
- Runtime errors occur while an application is running where the environment detects an operation that is impossible to carry out.
- For instance, an input error occurs when the user enters an unexpected input value that the program can’t handle. To prevent input errors, the program should prompt the user to enter the correct type of values.
- Another example of a run time error is division by zero.

```java
public class ShowRuntimeErrors {
    public static void main(String[] args) {
        int i = 1 / 0;
    }
}
```

1.10.3 Logic Errors

- Logic errors occur when a program doesn’t perform the way it was intended to.
- For example, the program doesn’t have syntax or runtime errors, but it does not print the correct result.

```java
// Suppose you wrote the following program to add number1 to number2
public class ShowLogicErrors {
    public static void main(String[] args) {
        // Add number1 to number2
        int number1 = 3;
        int number2 = 3;
        number2 += number1 + number2;
        System.out.println("number2 is "+number2);
    }
}
```