About the Presentations

- The presentations cover the objectives found in the opening of each chapter
- All chapter objectives are listed in the beginning of each presentation
- You may customize the presentations to fit your class needs
- Some figures from the chapters are included; a complete set of images from the book can be found on the Instructor Resources disc

Java Programming :Introduction

An Overview of Computers and Programming Languages

Chapter Objectives

- Learn about different types of computers
- Explore the hardware and software components of a computer system
- Learn about the language of a computer
- Learn about the evolution of programming languages
- Examine high-level programming languages

Chapter Objectives (continued)

- Discover what a compiler is and what it does
- Examine how a Java program is processed
- Learn what an algorithm is and explore problem-solving techniques
- Become aware of structured and object-oriented programming design methodologies

Introduction

- Computers have greatly affected our daily lives – helping us complete many tasks
- Computer programs (software) are designed specifically for each task
- Software is created with programming languages
- Java is an example of a programming language

An Overview of the History of Computers

- The first device known to carry out calculations was the abacus
- The abacus uses a system of sliding beads on a rack for addition and subtraction
- Blaise Pascal invented the calculating device called the Pascaline

Hardware Components of a Computer

- Central processing unit (CPU)
- Main memory

Central Processing Unit

- Arithmetic and logical operations are carried out inside the CPU

Central Processing Unit and Main Memory

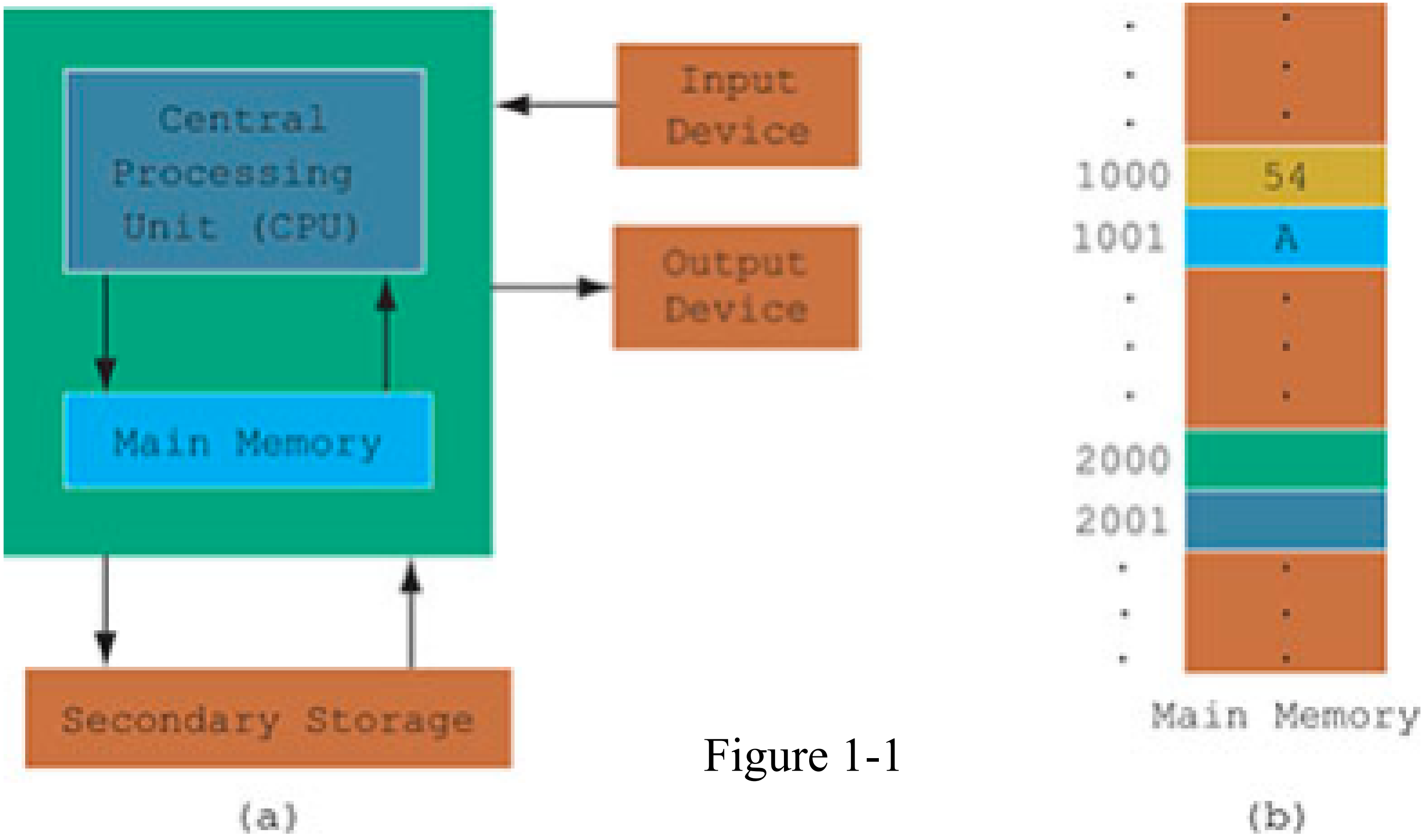


Figure 1-1

Main Memory

- Ordered sequence of cells (memory cells)
- Directly connected to CPU
- All programs must be brought into main memory before execution
- When power is turned off, everything in main memory is lost

Secondary Storage

- Provides permanent storage for information
- Examples of secondary storage:
  - Hard disks
  - Floppy disks
  - Flash memory
  - ZIP disks
  - CD-ROMs
  - Tapes

Input Devices

- Definition: devices that feed data and computer programs into computers
- Examples
  - Keyboard
  - Mouse
  - Secondary storage

Output Devices

- Definition: devices that the computer uses to display results
- Examples
  - Printer
  - Monitor
  - Secondary storage

Software

- Software consists of programs written to perform specific tasks
- Two types of programs
  - System programs
  - Application programs

System Programs

- System programs control the computer
- The operating system is first to load when you turn on a computer

Operating System (OS)

- OS monitors overall activity of the computer and provides services
- Example services
  - Memory management
  - Input/output
  - Activities
  - Storage management

Application Programs

- Written using programming languages
- Perform a specific task
- Run by the OS
- Example programs
  - Word processors
  - Spreadsheets
  - Games

Language of a Computer

- Machine language: the most basic language of a computer
- A sequence of 0s and 1s
- Every computer directly understands its own machine language
- A bit is a binary digit, 0 or 1
- A byte is a sequence of eight bits

Language of a Computer (continued)

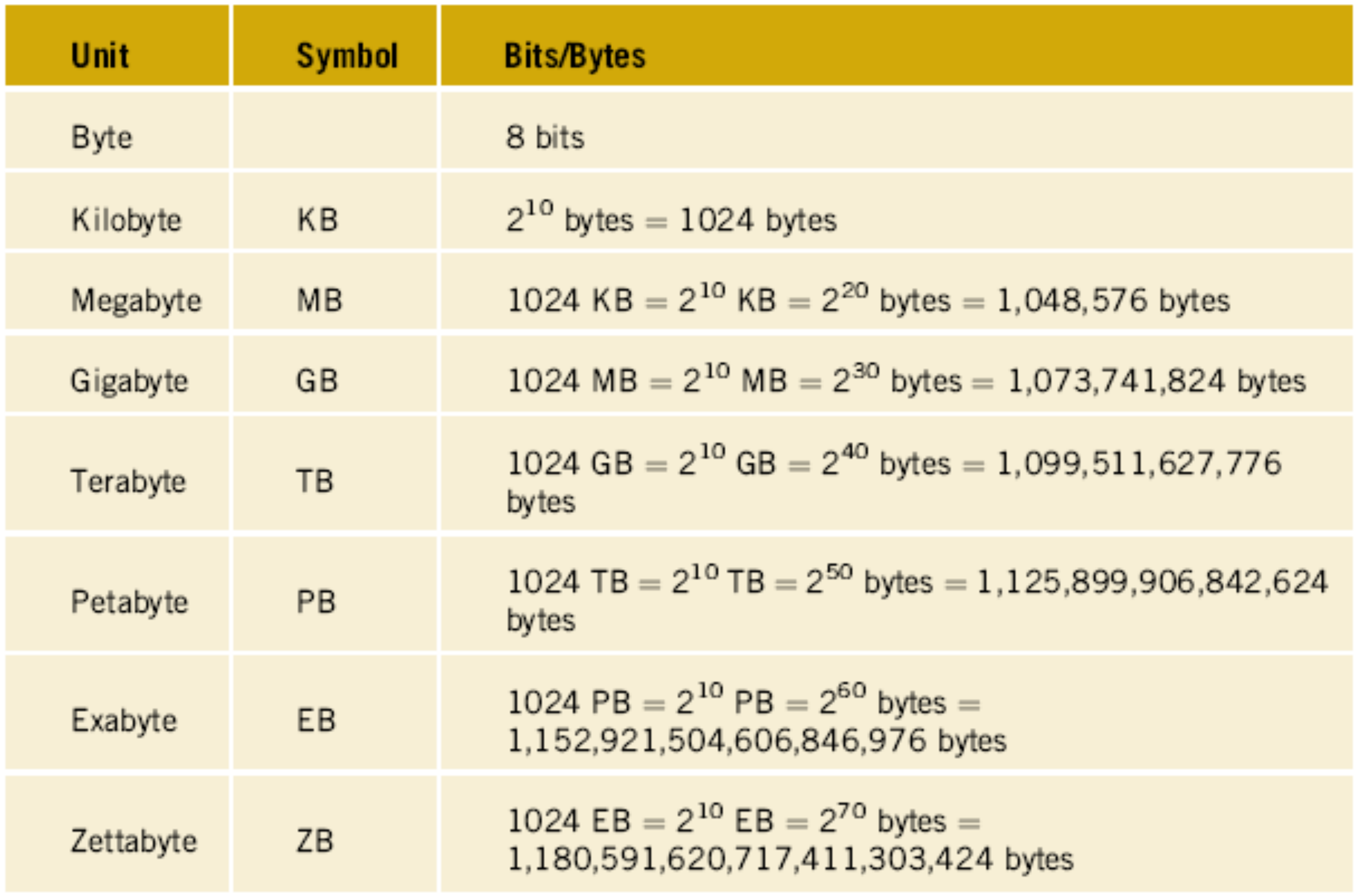


TABLE 1-1 Binary Units

Unit	Symbol	Bits/Bytes
Byte		8 bits
Kilobyte	KB	2^{10} bytes = 1024 bytes
Megabyte	MB	1024 KB = 2^{10} KB = 2^{20} bytes = 1,048,576 bytes
Gigabyte	GB	1024 MB = 2^{10} MB = 2^{30} bytes = 1,073,741,824 bytes
Terabyte	TB	1024 GB = 2^{10} GB = 2^{40} bytes = 1,099,511,627,776 bytes
Petabyte	PB	1024 TB = 2^{10} TB = 2^{50} bytes = 1,125,899,906,842,624 bytes
Exabyte	EB	1024 PB = 2^{10} PB = 2^{60} bytes = 1,152,921,504,606,846,976 bytes
Zettabyte	ZB	1024 EB = 2^{10} EB = 2^{70} bytes = 1,180,591,620,717,411,303,424 bytes

Evolution of Programming Languages

- Early computers programmed in machine language
- Assembly languages were developed to make programmer’s job easier
- In assembly language, an instruction is an easy-to-remember form called a mnemonic
- **Assembler**: translates assembly language instructions into machine language

Instructions in Assembly and Machine Language

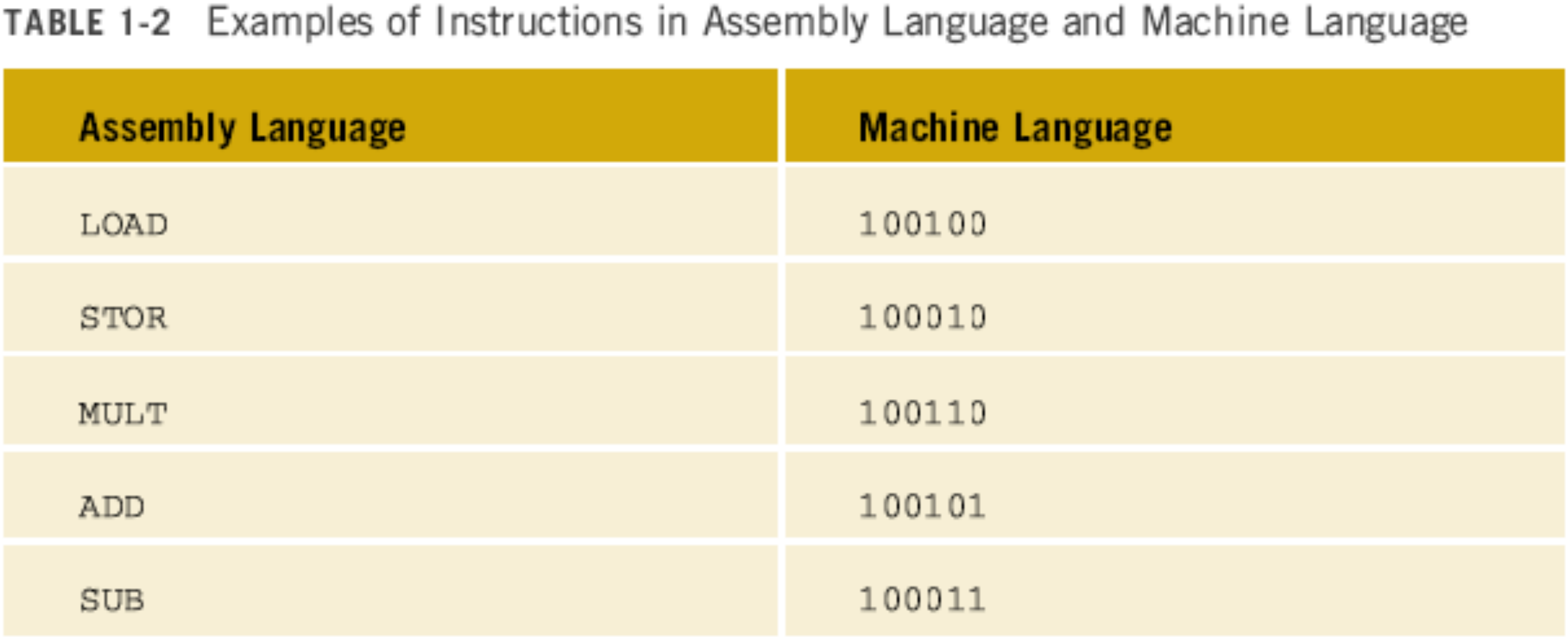


TABLE 1-2 Examples of Instructions in Assembly Language and Machine Language

Assembly Language	Machine Language
LOAD	100100
STOR	100010
MULT	100110
ADD	100101
SUB	100011

Evolution of Programming Languages

- High-level languages make programming easier
- Closer to spoken languages
- Examples
  - Basic
  - FORTRAN
  - COBOL
  - C/C++
  - Java

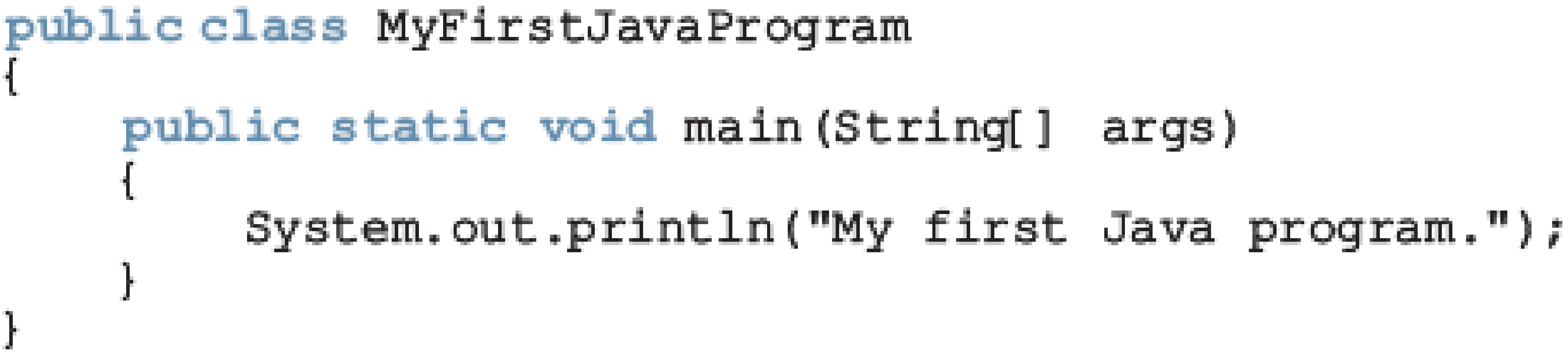
Evolution of Programming Languages (continued)

- To run a Java program:
  1. Java instructions need to be translated into an intermediate language called bytecode
  2. Then the bytecode is interpreted into a particular machine language

Evolution of Programming Languages (continued)

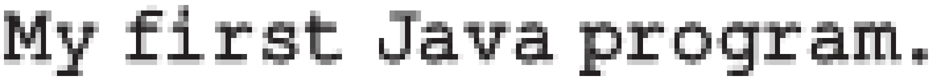
- **Compiler**: a program that translates a program written in a high-level language into the equivalent machine language
  - In the case of Java, this machine language is the bytecode
- **Java Virtual Machine (JVM)**: hypothetical computer developed to make Java programs machine independent

A Java Program



```
public class MyFirstJavaProgram
{
    public static void main(String[] args)
    {
        System.out.println("My first Java program.");
    }
}
```

Output:



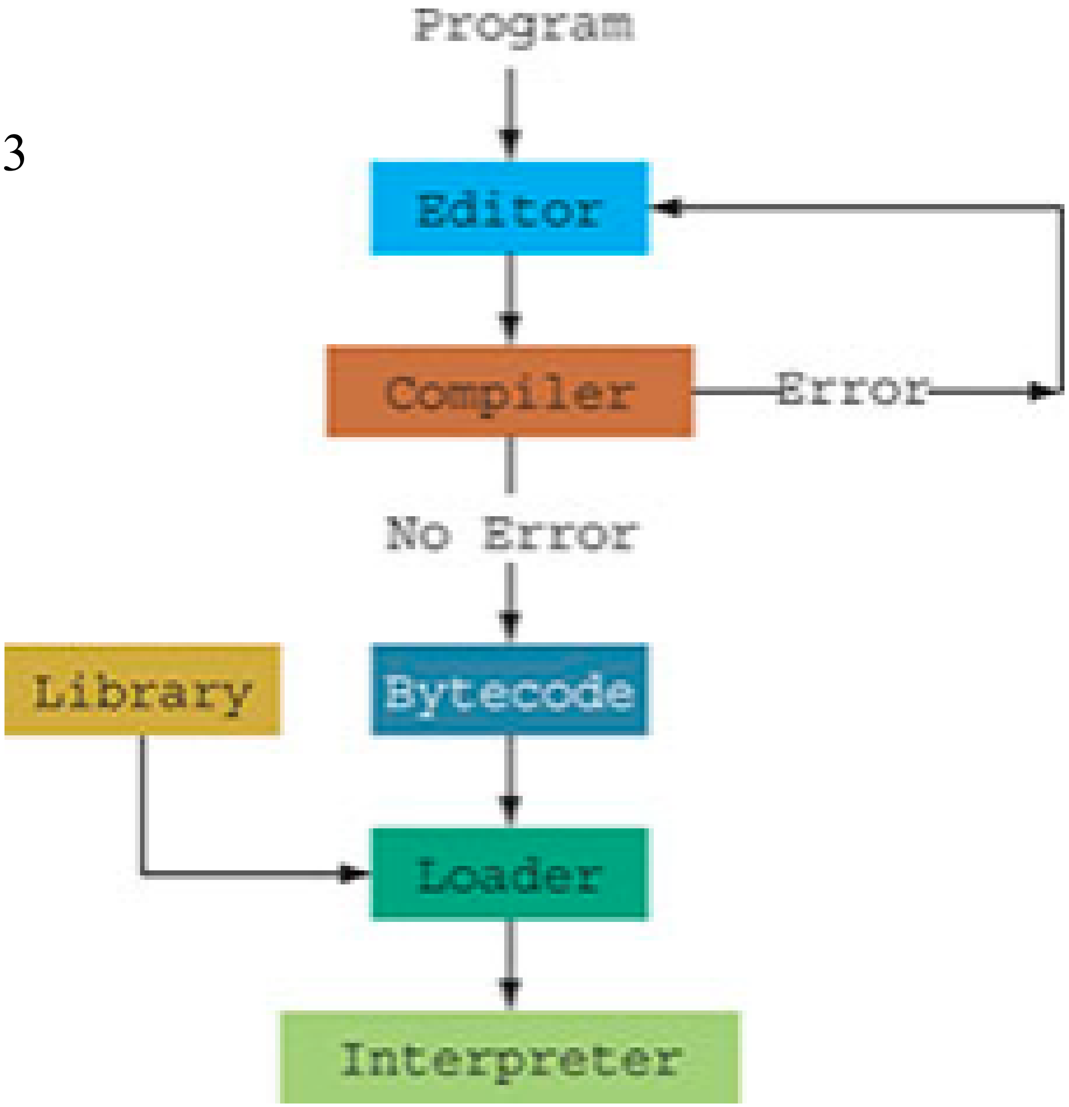
```
My first Java program.
```

Processing a Java Program

- Two types of Java programs: applications and applets
- **Source program**: written in a high-level language
- **Loader**: transfers the compiled code (bytecode) into main memory
- **Interpreter**: reads and translates each bytecode instruction into machine language and then executes it

Processing a Java Program (continued)

Figure 1-3



Internet, World Wide Web, Browser, and Java

- The *Internet* is an interconnection of networks that allows computers around the world to communicate with each other
- In 1969, the U.S. Department of Defense’s Advanced Research Project Agency (ARPA) funded research projects to investigate and develop techniques and technologies to interlink networks
- This was called the *internetting* project, and the funding resulted in ARPANET, which eventually became known as the “Internet”
- The Internet allows computers to be connected and communicate with each other

Internet, World Wide Web, Browser, and Java (continued)

- *World Wide Web* (WWW), or Web, uses software programs that enable computer users to access documents and files (including images, audio, and video) on almost any subject over the Internet with the click of a mouse
- Computers around the world communicate via the Internet; the World Wide Web makes that communication a fun activity

Internet, World Wide Web, Browser, and Java (continued)

- The primary language for the Web is known as *Hypertext Markup Language* (HTML)
- Java applets are programs that run from a *Web browser* and make the Web responsive and interactive
- Two well-known *browsers* are Mozilla Firefox and Internet Explorer
- Java applets can run in either browser
- Through the use of applets, the Web becomes responsive, interactive, and fun to use

Problem-Analysis-Coding-Execution Cycle

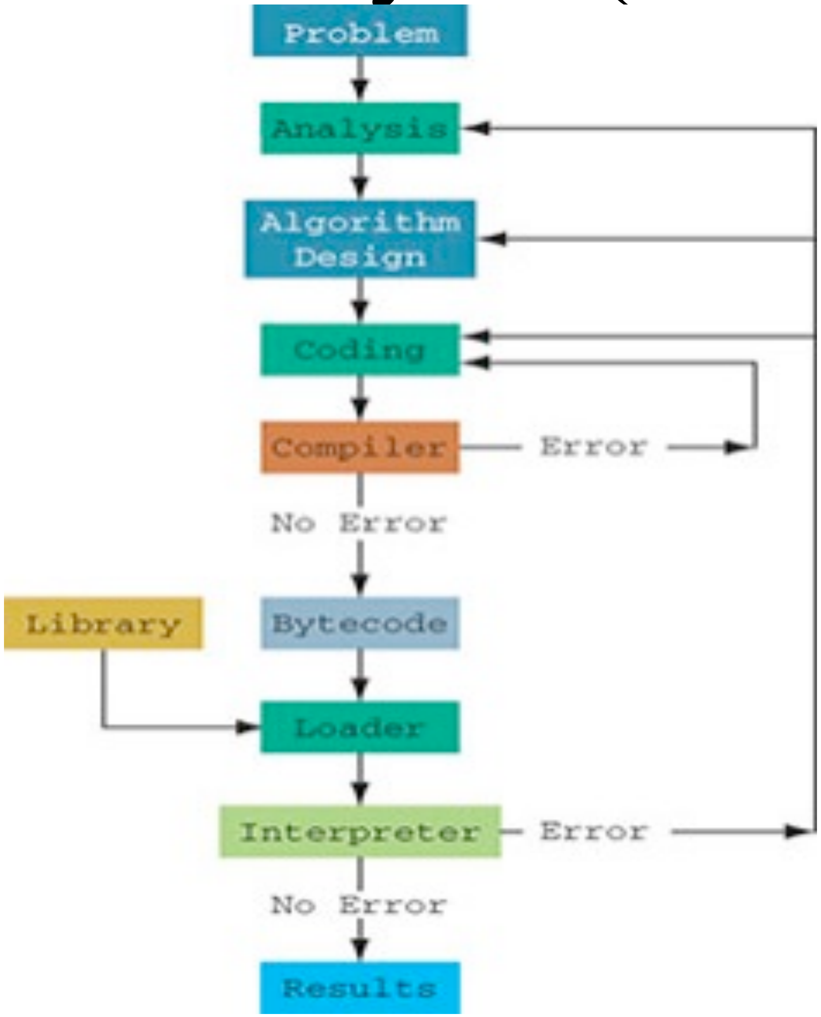
- **Algorithm**: a step-by-step problem-solving process in which a solution is arrived at in a finite amount of time

Problem-Solving Process

1. Analyze the problem and outline the problem and its solution requirements
2. Design an algorithm to solve the problem
3. Implement the algorithm in a programming language, such as Java
4. Verify that the algorithm works
5. Maintain the program by using and improving it and modifying it if the problem domain changes

Problem-Analysis-Coding-Execution Cycle (continued)

Figure 1-4



Programming Methodologies

- Two basic approaches to programming design
  - Structured design
  - Object-oriented design

Structured Design

1. A problem is divided into smaller subproblems
2. Each subproblem is solved
3. The solutions of all subproblems are then combined to solve the problem

Object-Oriented Design (OOD)

- In OOD, a program is a collection of interacting objects
- An object consists of data and operations
- Steps in OOD
  1. Identify objects
  2. Form the basis of the solution
  3. Determine how these objects interact

Chapter Summary

- A computer system is made up of hardware and software components
- Computers understand machine language; it is easiest for programmers to write in high-level languages
- A compiler translates high-level language into machine language
- Java steps to execute a program: edit, compile, load, and execute

Chapter Summary (continued)

- Algorithm: step-by-step problem-solving process in which a solution is arrived at in a finite amount of time
- Three steps to problem solving: analyze the problem and design an algorithm, implement the algorithm in a programming language, and maintain the program
- Two basic approaches to programming design: structured and object-oriented