Chapter 1 Background

Fundamentals of Java

Objectives

- Give a brief history of computers.
- Describe how hardware and software make up computer architecture.
- Understand the binary representation of data and programs in computers.

Objectives (cont.)

- Discuss the evolution of programming languages.
- Describe the software development process.
- Discuss the fundamental concepts of objectoriented programming.

Vocabulary

- Application software
- Bit
- Byte
- Central processing unit (CPU)
- Hardware
- Information hiding
- Object-oriented programming

Vocabulary (cont.)

- Primary memory
- Secondary memory
- Software
- Software development life cycle (SDLC)
- System software
- Ubiquitous computing
- Waterfall model

History of Computers

- 1940s: ENIAC (One of world's first digital electronic computers)
- 1950s: IBM sells first business computers
- 1960s: Time-sharing computers
- 1970s: Networking takes hold

History of Computers (cont.)

- 1980s: Many PCs, LANs become popular
- 1990s: Explosion in computer use
 - Internet becomes prevalent
- 21st century: Ubiquitous computing

Computer Hardware and Software

- Hardware: Physical devices that you see on your desktop
- Software: Programs that give the hardware useful functionality

Bits and Bytes

- Bit (binary digit): Smallest unit of information processed by a computer
 - A single 0 or 1
- Byte: 8 adjacent bits
- Capacity of computer memory and storage devices usually expressed in bytes

Bits and Bytes (cont.)

UNIT OF BYTES	NUMBER OF BYTES	TYPE OF STORAGE
Kilobyte	1000 bytes	A single file
Megabyte	1 million bytes	Large files, RAM, flash memory, CDs
Gigabyte	1 billion bytes	RAM, hard disk drives, DVDs
Terabyte	1000 gigabytes	File server disks

Table 1-1: Some commonly used quantities of information storage

Computer Hardware

- Six major sub-systems:
 - User interface: Supports moment-to-moment communication between user and computer
 - Keyboard and mouse
 - Auxiliary I/O devices: Printers and scanners
 - Auxiliary storage devices: Secondary memory
 - Hard disks, CD-ROMS, flash memory sticks

Computer Hardware (cont.)

- Six major sub-systems (continued):
 - Network connection: Connection to Internet
 - Modems
 - Internal memory: Random access memory (RAM) or primary memory
 - Fast and relatively small
 - Central processing unit (CPU): Performs the work on a computer
 - Consists of billions of transistors

Computer Software: System Software

- Supports basic operations of a computer and allows human interaction
 - Operating system
 - Communications software
 - Compilers
 - User interface subsystem

Computer Software: Application Software

- Allows human users to accomplish specialized tasks
 - Word processors
 - Spreadsheets
 - Database systems
 - Multimedia software

Binary Representation of Information & Computer Memory

- Computer memory stores patterns of electronic signals.
 - CPU reads, manipulates, and transforms patterns.
 - Patterns can be stored/viewed as strings of bits.
 - Sequences of 1s and 0s
 - To determine what a sequence of bits represents, you must know the context.

Integers

Computers use binary (base 2) notation.

$$-10011_2 = (1 * 2^4) + (0 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0)$$

 Computer scientists use binary, octal (base 8), and hexadecimal (base 16) notation.

Integers (cont.)

BASE 10	BASE 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
43	101011

Table 1-2: Some base 10 numbers and their base 2 equivalents

Floating-Point Numbers

- Numbers with a fractional part
- Mantissa/exponent notation: Number is rewritten as a value between 0 and 1 times a power of 10.
 - 354.9810 = 0.3549810 * 103
- IEEE standard: Mantissa contains one digit before the decimal point.

Characters and Strings

- ASCII (American Standard Code for Information Interchange) encoding scheme: Each character represented as a pattern of 8 bits (1 byte)
 - 256 characters may be represented
- Java uses Unicode encoding scheme:
 - 2 bytes used to represent a character
 - 65,536 characters may be represented

Characters and Strings (cont.)

CHARACTER	BIT PATTERN	CHARACTER	BIT PATTERN	CHARACTER	BIT PATTERN
A	0100 0001	а	0110 0001	0	0011 0000
В	0100 0010	b	0110 0010	1	0011 0001
		* * *		(a.ta.a.t	
Z	0101 1010	Z	0111 1010	9	0011 1001

Table 1-3: Some characters and their corresponding ASCII bit patterns

Sound

- Information contained in sound is analog.
 - Continuous waveform
- In order to represent in a computer, sound must be digitized.
 - Sampled at regular intervals on the waveform
 - Standard sampling rate is 44,000 samples per second
 - Requires large amount of storage

Sound (cont.)

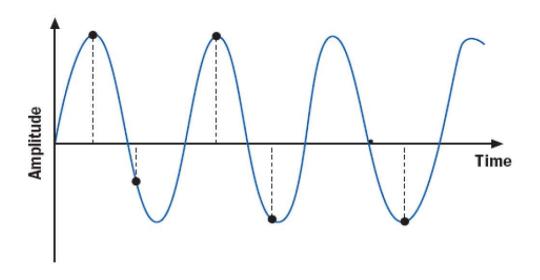


Figure 1-4a: Sampling a waveform

Images

- Also analog information
 - Set of color and intensity values spread across a two-dimensional space
 - Sampling devices: Scanners and digital cameras
 - Measure discrete values at pixels on a grid
 - Black-and-white: 2 bits per pixel
 - Grayscale: 8 bits per pixel for 256 shades of gray
 - True color (RGB): 24 bits per pixel

Video

- Consists of a soundtrack and frames
 - Sets of images recorded in sequence during a given time interval
- Primary challenge in digitizing video is data compression

Program Instructions

- Represented as a sequence of bits in RAM
- Example:
 - 0000 1001 / 0100 0000 / 0100 0010 / 0100 0100
 - First 8 bits represent the ADD command
 - Operation code or opcode
 - Second 8 bits represent first operand
 - Third 8 bits represent second operand
 - Fourth 8 bits used to store sum

Computer Memory

- Address: A byte's location in memory
 - Numbered from 0 to 1 less than the number of bytes of memory installed on the computer
 - Adjacent bytes may store different types of data.
 - Depends on context

Computer Memory (cont.)

Figure 1-5: A 32MB RAM

Address	Memory		
0			
1 2			
3	0100 1000		
4	0110 1001		
	*		
	•		
M ± 2 M ± 1			

32M ± 2 32M ± 1

Programming Languages

- Generation 1—machine languages:
 Program data entered directly into RAM in form of 1s and 0s
 - Using switches and, later, punch cards
 - Error prone, tedious, and slow

Programming Languages (cont.)

- Generation 2—assembly languages:
 Mnemonic symbols represent instructions and data.
 - One-to-one correspondence with machinelanguage instructions
 - Assembler: Translates to machine language
 - Loader: Loads machine language into memory

Programming Languages (cont.)

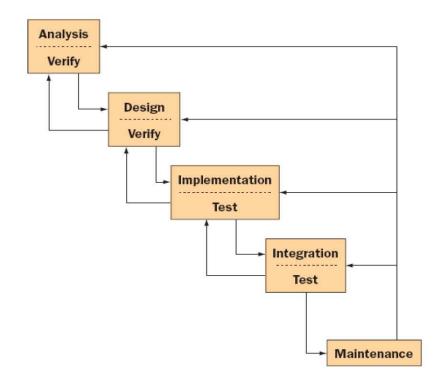
- Generation 3—high-level languages:
 Designed to be easy to write, read, and manipulate.
 - C, C++, Java
 - High-level instructions may represent many machine-language instructions.
 - Compiler translates high-level language into machine language.

The Software Development Process

- Software development life cycle (SDLC): A view of software development in which phases of development occur incrementally
- Standardizes software development
 - Simplifies understanding the project scope
 - Minimizes software flaws

- Waterfall model: A version of the SDLC
 - Phases:
 - Customer request
 - Analysis
 - Design
 - Implementation
 - Integration
 - Maintenance

Figure 1-6: Waterfall model of the software development life cycle



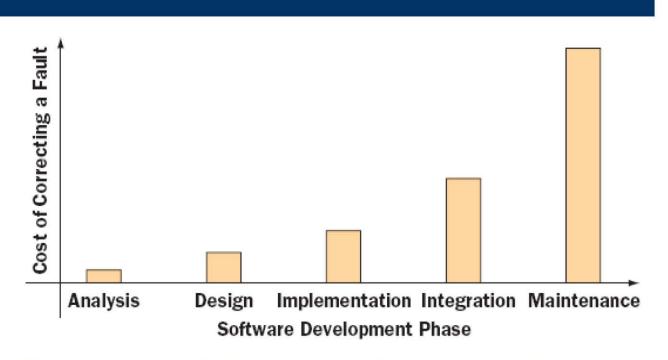


Figure 1-7: Relative costs of repairing mistakes when found in different phases

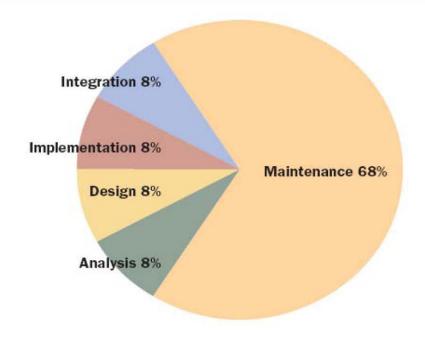


Figure 1-8: Percentage of total cost incurred in each phase of the development process

Basic Concepts of Object-Oriented Programming

- Two major groups of high-level languages:
 - Procedural languages: COBOL, FORTRAN, BASIC, C, Pascal
 - Object-oriented languages: Smalltalk, C++, Java
- Object-oriented approach is superior

Basic Concepts of Object-Oriented Programming (cont.)

- Object-oriented programming takes a divideand-conquer approach to reduce code.
 - Code can easily be reused
- A program consists of different types of software components called classes.
 - Defines data resources and methods
 - Rules of behavior

Basic Concepts of Object-Oriented Programming (cont.)

- Encapsulation: Combining the description of resources and behaviors into a single software entity
- A program is written first and then run.
- A running program is composed of interacting objects.
 - Each object's data resources (instance variables) and rules of behavior (methods) are defined by a class.

Basic Concepts of Object-Oriented Programming (Cont.)

- While a program is executing, it instantiates (creates) objects as needed.
- Objects work together to accomplish the mission of the program.
 - Send each other messages to request services or data
- When an object receives a message, it refers to its class to find the corresponding method to execute.

Basic Concepts of Object-Oriented Programming (cont.)

- Information hiding: Providing access to services, but not to data resources
- Classes are organized in a hierarchy.
- Root class defines methods and instance variables shared by its subclasses.
 - Those below it in the hierarchy
- Subclasses define additional methods and instance variables.

Basic Concepts of Object-Oriented Programming (cont.)

- Inheritance: The hierarchical class structure and sharing of instance variables and methods with subclasses
- Polymorphism: Different types of objects can understand the same message.
 - An object's response to a message depends on the object's class.

Summary

 The modern computer age began in the late 1940s with the development of ENIAC. Business computing became practical in the 1950s, and time-sharing computers advanced computing in large organizations in the 1960s and 1970s. The 1980s saw the development and first widespread sales of personal computers, and the 1990s saw personal computers connected in networks.

Summary (cont.)

 Modern computers consist of two primary components: hardware and software.
 Computer hardware is the physical component of the system. Computer software consists of programs that enable us to use the hardware.

Summary (cont.)

- All information used by a computer is represented in binary form. This information includes numbers, text, images, sound, and program instructions.
- Programming languages have been developed in the course of three generations: generation 1 is machine language, generation 2 is assembly language, and generation 3 is high-level language.

Summary (cont.)

- The software development process consists of several standard phases: customer request, analysis, design, implementation, integration, and maintenance.
- Object-oriented programming is a style of programming that can lead to better quality software. Breaking code into easily handled components simplifies the job of writing a large program.