Chapter 1 – Introduction to Computers, Internet and Visual Basic .NET

Outline

1.1 Introduction
1.2 What is a Computer?
1.3 Computer Organization
1.4 Evolution of Operating Systems
1.5 Personal Computing, Distributed Computing and Client/Server Computing
1.6 Machine Languages, Assembly Languages and High-Level Languages
1.7 Visual Basic .NET
1.8 C, C++, Java and C#
1.9 Other High-Level Languages
1.10 Structured Programming
1.11 Key Software Trend: Object Technology
1.12 Hardware Trends
1.13 History of the Internet and World Wide Web
1.14 World Wide Web Consortium (W3C)
1.15 Extensible Markup Language (XML)

© 2002 Prentice Hall. All rights reserved.

◀ ▶

Chapter 1 – Introduction to Computers, Internet and Visual Basic .NET

Outline

1.16 Introduction to Microsoft .NET

1.17 .NET Framework and the Common Language Runtime

© 2002 Prentice Hall. All rights reserved.

◀ ▶

1.1 Introduction

© 2002 Prentice Hall. All rights reserved.

1.2 What is a Computer?

- Computer
  - Device capable of
    - Performing computations
    - Making logical decisions
  - Works billions of times faster than human beings
  - Fastest *supercomputers* today
    - Perform hundreds of billions of additions per second

© 2002 Prentice Hall. All rights reserved.

1.2 What is a Computer? (II)

- Programs
  - Sets of instructions that process data
  - Guide computer through orderly sets of actions specified by *computer programmers*
- Computer system
  - Comprised of various *hardware* devices
    - Keyboard
    - Screen (monitor)
    - Disks
    - Memory
    - Processing Units

© 2002 Prentice Hall. All rights reserved.

1.3 Computer Organization

- Every computer divided into six units

  1. Input unit

  - “Receiving” section of computer
  - Obtains data from *input devices*
    - Usually a keyboard, mouse, disk or scanner
  - Places data at disposal of other units

  2. Output unit

  - “Shipping” section of computer
  - Puts processed info on various *output devices*
    - Screens, paper printouts, speakers
  - Makes info available outside the computer

© 2002 Prentice Hall. All rights reserved.

1.3 Computer Organization (II)

3. Memory unit

- Rapid access, low capacity “warehouse”
- Retains information entered through input unit
- Retains info that has already been processed until can be sent to output unit
- Often called *memory*, *primary memory*, or ***r**andom **a**ccess **m**emory* (RAM)

*4. **A**rithmetic and **L**ogic **U**nit*

- “Manufacturing” section of computer
- Performs calculations (addition, subtraction, multiplication and division)
- Contains decision mechanisms and can make comparisons

© 2002 Prentice Hall. All rights reserved.

1.3 Computer Organization (III)

5. *Central **P**rocessing **U**nit* (CPU)
   - "Administrative" section of computer
   - Coordinates and supervises other sections
6. *Secondary storage unit*
   - Long-term, high-capacity "warehouse"
   - Stores programs or data not currently being used by other units on *secondary storage devices* (like discs)
   - Takes longer to access than primary memory

© 2002 Prentice Hall. All rights reserved.

1.4 Evolution of Operating Systems

- Early Computers
  - Single-user batch processing
    - Jobs on decks of punched cards
    - One job ran at a time
    - Results took hours to process
  - Operating Systems
    - Managed transitions between jobs
    - Increased amount of work computer could accomplish
  - Multiprogramming
    - Simultaneous operation of several jobs
    - Computer resources split between jobs
    - Still took long hours for results

© 2002 Prentice Hall. All rights reserved.

◀ ▶

1.4 Evolution of Operating Systems (II)

- Timesharing Operating Systems (1960s)
  - Computers accessed through terminals
    - Devices with keyboards and screens
  - Hundreds of people use system at once
    - Quickly performs small portions of each person's job
    - Gives appearance of running simultaneously

© 2002 Prentice Hall. All rights reserved.

1.4 Evolution of Operating Systems (III)

- UNIX
  - Originally an experimental timesharing OS
  - Developed by Dennis Ritchie and Ken Thompson
    - Developed at Bell Labs
    - Written in C
  - Open-Source software
    - Source code freely distributed among programmers
    - Created large community
  - Powerful and flexible
    - Handled any task a user required
    - Developed into many versions
      - Linux
      - BSD

© 2002 Prentice Hall. All rights reserved.

◀ ▶

1.5 Personal Computing, Distributed Computing and Client/Server Computing

- Personal Computing
  - Pioneered by Apple and IBM
    - Computer prices dropped
      - Computers reasonable for personal or business use
  - Stand-alone units
    - People work on personal machines
    - Transported disks to share information
      - “Sneakernet”
    - Machines linked together
      - Telephone lines
      - Local Area Networks (LANs)
      - Led to distributed computing

© 2002 Prentice Hall. All rights reserved.

◀ ▶

1.5 Personal Computing, Distributed Computing and Client/Server Computing (II)

- Distributed Computing
  - Work distributed over networks
    - N-Tier applications
      - Split parts of applications over numerous computers
        - User interface
        - Database
        - Business-logic processing
      - Different parts interact when application runs

© 2002 Prentice Hall. All rights reserved.

1.5 Personal Computing, Distributed Computing and Client/Server Computing (III)

- Client/Server Computing
  - Workstations
    - High-powered desktop machines
    - Easily share information over computer networks
  - Servers
    - Store programs and data
    - Information accessed by clients
  - Capabilities provided by modern Operating Systems
    - Windows (and its variants), UNIX, Linux, MacOS

© 2002 Prentice Hall. All rights reserved.

1.6 Machine Languages, Assembly Languages and High-Level Languages

- Programming Languages
  - Hundreds exist today
  - Fall into three categories
    - Machine languages
    - Assembly languages
    - High-level languages

© 2002 Prentice Hall. All rights reserved.

1.6 Machine Languages, Assembly Languages and High-Level Languages (II)

- Machine Languages
  - Only language understood directly by computer
  - Defined by computer’s hardware design
    - Machine-dependent
      - Languages specific to particular computers
  - Incomprehensible to human readers
    - Streams and numbers
      - Ultimately reduced to 0s and 1s
      - Instruct most elementary of operations
    - Slow, tedious and error-prone
      - Led to Assembly languages

© 2002 Prentice Hall. All rights reserved.

1.6 Machine Languages, Assembly Languages and High-Level Languages (III)

- Assembly Languages
  - English-like abbreviations
    - Represent elementary operations of computer
  - Translated to machine language
    - Assemblers convert to machine language
    - High-speed conversion
  - More clear to human readers
    - Still tedious to use
      - Many instructions for simple tasks
      - Led to high-level languages

© 2002 Prentice Hall. All rights reserved.

1.6 Machine Languages, Assembly Languages and High-Level Languages (IV)

- High-Level Languages
  - Single statements accomplish substantial tasks
  - Translated to machine language
    - Compilers convert to machine language
    - Conversion takes considerable time
      - Interpreters run programs without compiling
        - Used in development environment
  - Instructions comprehensible to humans
    - Look like everyday English
    - Contain common mathematical notation

© 2002 Prentice Hall. All rights reserved.

1.7 Visual Basic .NET

- History of Visual Basic
  - Evolved from BASIC
    - Beginner's All-Purpose Symbolic Instruction Code
    - Introduction to programming for novices
  - Introduced by Microsoft (1991)
    - Made programming for Windows much easier
      - IDE facilitates quicker programming
        - Rapid Application Development
      - Provided many extra features
        - Event handling
        - Graphical user interfaces
        - Object-oriented programming

© 2002 Prentice Hall. All rights reserved.

1.7 Visual Basic .NET (II)

- .NET Strategy
  - Motivated by popularity of electronic devices
  - Provide interaction between wide variety of devices
    - Web-based applications distributed to different devices
      - Cell phones
      - Personal Digital Assistants
      - Personal computers
    - Allow communication between programs of disparate languages

© 2002 Prentice Hall. All rights reserved.

1.7 Visual Basic .NET (III)

- Visual Basic .NET
  - Enhanced object orientation
    - Powerful library of components
      - Application development made even quicker
  - Enhanced language interoperability
    - Interaction between different languages
      - Old software compatible with Visual Basic .NET programs
      - Interaction over the Internet
        - Simple Object Access Protocol (SOAP)
        - Extensible Markup Language (XML)
  - Promotes use of Web services

© 2002 Prentice Hall. All rights reserved.

1.8 C, C++, Java and C#

- History of C
  - Developed by Dennis Ritchie
    - Evolved from the B language at Bell Labs
    - Added data type features
  - Gained recognition as language of UNIX
  - Now a widely used language
    - Available on most computers
    - Language of most major operating systems
    - Led to development of C++

© 2002 Prentice Hall. All rights reserved.

1.8 C, C++, Java and C# (II)

- C++
  - Developed by Bjarne Stroustrup at Bell Labs (1980s)
  - Extension of C
    - Uses elements from Simula 67
      - Simulation programming language
    - Provides features to “spruce up” C
    - Provides Object-oriented technology
      - Hybrid language
        - Possible to program structurally
        - Possible to program with object-oriented technology
        - Possible to use both

© 2002 Prentice Hall. All rights reserved.

1.8 C, C++, Java and C# (III)

- Object-Oriented Programming
  - Uses objects
    - Reusable software components
    - Modeled after items in the real world
  - More productive than structured programming
  - Easier to understand, correct and modify

© 2002 Prentice Hall. All rights reserved.

1.8 C, C++, Java and C# (IV)

- History of Java
  - Sun Microsystems corporate research project (1991)
    - Code-name Green
  - Based on C and C++
  - Intended for intelligent consumer-electronic devices
    - Lack of popularity almost causes cancellation
  - Sudden popularity of WWW provides new potential
    - Java capable of dynamic content
      - Animated and interactive content
    - Grabbed attention of business community
  - Now very widely used
    - Enhance functionality of WWW servers
    - Provide applications for consumer devices

© 2002 Prentice Hall. All rights reserved.

1.8 C, C++, Java and C# (V)

- C#
  - Developed by Anders Hejlsberg and Scott Wiltamuth at Microsoft
  - Roots in C, C++ and Java
    - Adapts best features of each language
  - Designed for .NET platform specifically
  - Provides many features associated with .NET
    - Object-oriented
    - Powerful class library of pre-built components

© 2002 Prentice Hall. All rights reserved.

1.9 Other High-Level Languages

- Fortran (Formula Translator)
  - Developed at IBM (1950s)
    - Scientific and engineering applications
    - Creates scientific and engineering applications
    - Still widely used
- COBOL (Common Business Oriented Language)
  - Developed by computer users, manufacturers and the government (1959)
    - Efficient manipulation of large amounts of data
    - Commonly associated with business software
- Pascal
  - Developed by Professor Nicklaus Wirth (late 1960s)
    - For academic use

© 2002 Prentice Hall. All rights reserved.

1.10 Structured Programming

- Early Software Development
  - Complex and costly for businesses (1960s)
    - Costs exceeded budgets
    - Final products unreliable
  - Research led to structured programming
    - Disciplined approach to programming
      - Programs clear and easy to modify
    - Several languages resulted from research
      - C, Pascal, Ada

© 2002 Prentice Hall. All rights reserved.

1.10 Structured Programming (II)

- Structured Languages
  - Pascal
    - Designed for teaching structured programming
    - Lacked features for commercial use
  - C
    - Had features Pascal didn’t
    - Quickly adopted by programmers
  - Ada
    - Developed by U.S. Department of Defense (late 1970s)
      - Based on Pascal
      - DOD wanted one language for all its needs
      - Supported multitasking
        - Many activities occur in parallel

© 2002 Prentice Hall. All rights reserved.

1.11 Key Software Trend: Object Technology

- What is Object Technology?
  - Packaging scheme for creating software units
    - Units are objects
      - Any noun can be represented as an object
        - Date object, time object, car object
      - Have properties
        - Size, color, weight
      - Perform actions
        - Moving, sleeping, drawing
      - Defined in classes
        - Specify general format
        - Provide specific attributes and behaviors

© 2002 Prentice Hall. All rights reserved.

1.11 Key Software Trend: Object Technology (II)

- Object-Oriented programming
  - Based on nouns
    - Reflects way world is perceived
  - Advantages over structured programming
    - More natural process
      - Results in better productivity
    - Classes provide reusability
      - Microsoft Foundation Classes (MFC)
    - Easier to maintain
      - Programs more understandable
      - Focus on behaviors and interactions
      - Less attention to details

© 2002 Prentice Hall. All rights reserved.

1.12 Hardware Trends

- Improving technologies
  - Internet community thrives on improvements of
    - Hardware
    - Software
    - Communications
  - Cost of products and services
    - Consistently dropping over the decades
  - Computer capacity and speed
    - Doubles every two years (on average)
  - Microprocessor chip
    - Laid groundwork in late 1970s and 1980s for productivity improvements of the 1990s

© 2002 Prentice Hall. All rights reserved.

1.13 History of the Internet and World Wide Web

- *ARPAnet*
  - Implemented in late 1960’s by ARPA (Advanced Research Projects Agency of DOD)
  - Networked computer systems of a dozen universities and institutions with 56KB communications lines
  - Grandparent of today’s Internet
  - Intended to allow computers to be shared
  - Became clear that key benefit was allowing fast communication between researchers – *electronic-mail* (*email*)

© 2002 Prentice Hall. All rights reserved.

1.13 History of the Internet and World Wide Web (II)

- ARPA’s goals
  - Allow multiple users to send and receive info at same time
  - Network operated *packet switching* technique
    - Digital data sent in small packages called *packets*
    - Packets contained data, address info, error-control info and sequencing info
    - Greatly reduced transmission costs of dedicated communications lines
  - Network designed to be operated without centralized control
    - If portion of network fails, remaining portions still able to route packets

© 2002 Prentice Hall. All rights reserved.

1.13 History of the Internet and World Wide Web (III)

- ***T**ransmission **C**ontrol **P**rotocol* (TCP)
  - Name of protocols for communicating over ARPAnet
  - Ensured that messages were properly routed and that they arrived intact
- Organizations implemented own networks
  - Used both for intra-organization and communication

© 2002 Prentice Hall. All rights reserved.

1.13 History of the Internet and World Wide Web (IV)

- Huge variety of networking hardware and software appeared
  - ARPA achieved inter-communication between all platforms with development of the *IP*
    - *Internetworking Protocol*
    - Current architecture of Internet
  - Combined set of protocols called *TCP/IP*
- The Internet
  - Limited to universities and research institutions
  - Military became big user
  - Next, government decided to access Internet for commercial purposes

© 2002 Prentice Hall. All rights reserved.

1.13 History of the Internet and World Wide Web (V)

- Internet traffic grew
  - Businesses spent heavily to improve Internet
    - Better service their clients
  - Fierce competition among communications carriers and hardware and software suppliers
  - Result
    - *Bandwidth* (info carrying capacity) of Internet increased tremendously
    - Costs plummeted

© 2002 Prentice Hall. All rights reserved.

1.13 History of the Internet and World Wide Web (VI)

- WWW
  - Allows computer users to locate and view multimedia-based documents
  - Introduced in 1990 by Tim Berners-Lee
    - Developed information system based on hyperlinked text documents
      - HyperText Markup Language (HTML)
      - Developed communication protocols as backbone
- WWW today
  - Makes information instantly accessible
  - Merges computing and communication technologies

© 2002 Prentice Hall. All rights reserved.

1.14 World Wide Web Consortium (W3C)

- W3C
  - Founded in 1994 by Tim Berners-Lee
    - Devoted to developing non-proprietary and interoperable technologies for the World Wide Web and making the Web universally accessible
  - Standardization
    - W3C *Recommendations*: technologies standardized by W3C
      - include Extensible HyperText Markup Language (XHTML), Cascading Style Sheets (CSS) and the Extensible Markup Language (XML)
    - Document must pass through *Working Draft, Candidate Recommendation* and *Proposed Recommendation* phases before considered for W3C Recommendation

© 2002 Prentice Hall. All rights reserved.

1.14 World Wide Web Consortium (W3C) (II)

- W3C Structure
  - 3 Hosts
    - Massachusetts Institute of Technology (MIT)
    - INRIA (Institut National de Recherche en Informatique et Automatique)
    - Keio University of Japan
  - 400 Members (including Deitel & Associates)
- W3C homepage at www.w3.org
- W3C Goals
  - User Interface Domain
  - Technology and Society Domain
  - Architecture Domain and Web Accessibility Initiatives

© 2002 Prentice Hall. All rights reserved.

1.15 Extensible Markup Language

- HTML limitations
  - Lack of extensibility
    - Inability to add or change features
      - Developers become frustrated
      - Code becomes erroneous
  - Led to more development on HTML
    - W3C created Cascading Style Sheets as temporary solution
      - New technology for formatting documents
    - Led to research for a standardized extensible language
      - W3C developed Extensible Markup Language (XML)
        - Combined power of SGML with simplicity of HTML
        - Developed XML-based standards for style-sheets and advanced hyperlinking

© 2002 Prentice Hall. All rights reserved.

1.15 Extensible Markup Language (II)

- XML features
  - Data independence
    - Separation of content from its presentation
      - Allows any application to conceivably process XML documents
      - Improves Web functionality and interoperability
      - Reduces server load and network traffic
    - Integration with applications other than Web services
      - Communication between applications employing XML
      - Structure allows easy integration with database applications

© 2002 Prentice Hall. All rights reserved.

1.15 Extensible Markup Language (III)

- Communication using XML
  - Simple Object Access Protocol (SOAP)
    - Framework for expressing application semantics, encoding that data and packing it in modules
    - Structured into three parts
      - Envelope
        - Describes content and recipient of SOAP message
      - Encoding rules
        - XML-based
      - Remote Procedure Call (RPC) representation
        - Commands other computers to perform a task

© 2002 Prentice Hall. All rights reserved.

1.16 Introduction to Microsoft .NET

- .NET initiative
  - Introduced by Microsoft (June 2000)
    - Vision for embracing the Internet in software development
  - Independence from specific language or platform
    - Applications developed in any .NET compatible language
      - Visual Basic .NET, Visual C++ .NET, C# and more
    - Programmers can contribute to applications using the language in which they are most competent
  - Architecture capable of existing on multiple platforms
  - New program development process
    - Provides increased productivity

© 2002 Prentice Hall. All rights reserved.

1.16 Introduction to Microsoft .NET (II)

- Key components of .NET
  - Web services
    - Applications used over the Internet
  - Software reusability
    - Web services provide solutions for wide variety of companies
      - Cheaper than developing one-time solutions that can’t be reused
      - Single applications perform all operations for a company via various Web services
        - Manage taxes, bills, investments and more
    - Pre-packaged components
      - Make application development quicker and easier
      - Developers no longer need to be concerned with details of components

© 2002 Prentice Hall. All rights reserved.

1.16 Introduction to Microsoft .NET (III)

- Keys to interaction
  - XML and SOAP
    - "Glue" that combines various Web services to form applications
      - XML gives meaning to data
      - SOAP allows communication to occur easily

© 2002 Prentice Hall. All rights reserved.

1.16 Introduction to Microsoft .NET (IV)

- Other concepts
  - Universal data access
    - Eliminates need to synchronize files
      - Updating multiple copies of same file to the most recent
    - Data resides at one central location
      - Accessible by anyone with connection and proper authorization
    - Data formatted appropriately for display on various devices
      - Same document seen on PC, PDA, cell phone and other devices
  - Additional information available at Microsoft Web site www.microsoft.com/net

© 2002 Prentice Hall. All rights reserved.

1.17 .NET Framework and the Common Language Runtime

- .NET Framework
  - Heart of .NET strategy
    - Manages and executes applications and Web services
    - Provides security, memory management and other programming capabilities
  - Includes Framework class library (FCL)
    - Pre-packaged classes ready for reuse
    - Used by any .NET language
  - Details contained in Common Language Specification (CLS)
    - Submitted to European Computer Manufacturers Association to make the framework easily converted to other platforms
  - Executes programs by Common Language Runtime (CLR)

© 2002 Prentice Hall. All rights reserved.

1.17 .NET Framework and the Common Language Runtime (II)

- Common Language Runtime (CLR)
  - Central part of framework
    - Executes Visual Basic .NET programs
  - Compilation process
    - Two compilations take place
      - Programs compiled to Microsoft Intermediate Language (MSIL)
        - Defines instructions for CLR
      - MSIL code translated into machine code
        - Machine code for a particular platform

© 2002 Prentice Hall. All rights reserved.

1.17 .NET Framework and the Common Language Runtime (III)

- Why two compilations?
  - Platform independence
    - .NET Framework can be installed on different platforms
    - Execute .NET programs without any modifications to code
  - Language independence
    - .NET programs not tied to particular language
    - Programs may consist of several .NET-compliant languages
    - Old and new components can be integrated

© 2002 Prentice Hall. All rights reserved.

1.17 .NET Framework and the Common Language Runtime (IV)

- Other advantages of CLR
  - Execution-management features
    - Manages memory, security and other features
      - Relieves programmer of many responsibilities
      - More concentration on program logic

© 2002 Prentice Hall. All rights reserved.