

1

Introduction to Computers and Visual Basic.Net



OBJECTIVES

In this chapter you will learn:

- Basic computing concepts.
- The different types of programming languages.
- The evolution of the Basic Programming Language
- What .Net is and its impact on both Graphical and Web User Interfaces.
- What Graphical User Interfaces are.



- 1.1 Introduction
- 1.2 What Is a Computer?
- 1.3 Computer Organization
- 1.4 Machine Languages, Assembly Languages and High-Level Languages
- 1.5 Personal, Distributed and Client/Server Computing
- 1.6 Hardware Trends
- 1.7 Key Software Trend: Object Technology
- 1.8 C, C++ and Java
- 1.9 BASIC, Visual Basic, Visual C++, C# and .NET
- 1.10 Software Technologies
- 1.11 Web Resources



1.1 Introduction (Cont.)

■ Software

- Instructions to command the computer to perform actions and make decisions)
- Visual Basic is a popular software development languages for windows-based applications and can also be used for web and mobile devices.

■ Computer development

- Computer use increasing in most fields
- Computer costs and size decreasing
 - Abundance of silicon drives down prices of silicon-chip technology

■ Applications of this book

- Prepares for higher learning in C++, Java, C#, Visual Basic.NET as well as object-oriented programming
- Allows development of applications with graphical user interfaces (GUIs)
 - Multimedia capabilities
 - Integration with the Internet and World Wide Web



1.1 Introduction (Cont.)

- **Apply database technologies**
- **Applications that are not limited to the desktop**
- **Book structure**
 - **Focus on Visual Basic.Net 2005/2008**
 - **Chapters 1-11**
 - **Cover introductory programming concepts, user interface, etc.**
 - **Appendix A-D**
 - **Cover special topics like database integration.**



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**



1.2 What is a Computer? (Cont.)

■ 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
 - Disks
 - Memory
 - DVD drives
 - Processing Units



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, scanner, uploads (photos and videos) and networks (Internet)

- 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 (e.g., Internet)



1.3 Computer Organization (Cont.)

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 random access memory (RAM)**

4. Arithmetic and Logic Unit

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



1.3 Computer Organization (Cont.)

5. Central Processing Unit (CPU)

- “Administrative” section of computer
- Coordinates and supervises other sections
- Multiple CPUs (multiprocessors)

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 CDs and DVDs)
- Takes longer to access than primary memory



1.4 Machine Languages, Assembly Languages and High-Level Languages

- **Three general types of programming languages**
 - **Machine languages**
 - **Assembly languages**
 - **High-level languages**



1.4 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

■ Machine languages

- “Natural language” of a computer (aka object code)
- Defined by hardware design of computer
- Generally consists of strings of numbers
- Are machine dependent
- Cumbersome for humans
 - Example: Adding overtime pay to base pay and storing the result in gross pay
 - +1300042774
 - +1400593419
 - +1200274027
- Slow and tedious for most programmers



1.4 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

■ Assembly languages

- Programmers began using English-like abbreviations to substitute for machine languages
- Represents elementary operations of computer
- Translator programs called assemblers convert assembly-language to machine-language
- Example:

LOAD BASEPAY

ADD OVERPAY

STORE GROSSPAY



1.4 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

■ High-level languages

- Developed as computer usage increased, assembly language proved inadequate and time-consuming
- Single statements can be written to accomplish substantial tasks
- Translator programs called compilers
- Allow programmers to write instructions almost like every-day English
- Example:

`grossPay = basePay + overTimePay`



1.4 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

▪ High-level languages (II)

- Much more desirable from programmer's standpoint
- Specific languages include
 - C, C++, Visual Basic.NET, C# and Java
 - Among most powerful and widely used languages today
- Interpreter programs developed to execute high-level programs without compiling
 - Popular in program development environments
- Once program developed, compiled version made
- Internet programming languages
 - JavaScript, ActionScript, PHP and Ruby on Rails—each of these scripting languages is processed by interpreters
- Markup languages
 - XHTML and XML, which can be processed by interpreted scripting languages
 - Achieve their goal of portability across a variety of platforms



1.5 Personal, Distributed and Client/Server Computing

- **1977 Apple Computer popularized personal computing**
 - Computers became economical for personal or business use
- **Machines could be linked together in computer networks**
 - Local area networks (LANs)
 - Distributed computing
- **Workstations**
- **Servers offer data storage and other capabilities that may be used by client computers distributed throughout the network,**
 - Client/server computing
- **Popular operating systems**
 - UNIX, Linux, Mac OS X and Microsoft's Windows



1.6 Hardware Trends

- **Improving technologies**
 - **Internet community thrives on improvements of**
 - **Hardware, Software and Communications**
 - **Cost of products and services**
 - **Consistently dropping over the decades**
 - **Computer capacity and speed**
 - **Doubles every two years (on average) = Moore's Law**
 - **Microprocessor chip**
 - **Laid groundwork in late 1970s and 1980s for productivity improvements of the 1990s**
 - **Hardware moving toward mobile, wireless technology.**
 - **Hand-held devices more powerful than early supercomputers**
 - **Portability**
 - **Wireless data-transfer speeds**



1.7 The Key Software Trend: Object Technology

■ Objects

- Reusable software components that model items in the real world (classes)
- Makes software developers more productive
- Object-oriented programs often easier to understand, correct and modify than older types of programs



1.7 The Key Software Trend: Object Technology (Cont.)

■ Object technology

- **Packaging scheme that helps create meaningful software units**
 - **Large and highly focused on particular applications areas**
- **Before appeared, programming languages were focused on actions (verbs) rather than on objects (nouns)**
 - **Programmers would program primarily with verbs**
 - **Made program awkward**
 - **We live in a world filled with complex objects and simple actions**



1.7 The Key Software Trend: Object Technology (Cont.)

- **Object technology (continued)**
 - **Object-oriented programming**
 - **Programmers work in manner similar to how they see the world**
 - **More natural process**
 - **Significant productivity enhancements**
 - **Procedural programming**
 - **Not particularly reusable**
 - **Forces programmers to constantly “re-invent the wheel”**
 Wastes time and resources
 - **Objects (classes)**
 - **Software modules**
 - **Kept in libraries**
 - **Reusable – save time and resources**



Software Engineering Observation 1.2

Extensive class libraries of reusable software components are available on the Internet. Many of these libraries are free.



Software Engineering Observation 1.3

Some organizations report that the key benefit object-oriented programming gives them is not software that is reusable but, rather, software that is more understandable, better organized and easier to maintain, modify and debug. This can be significant, because perhaps as much as 80 percent of software cost is associated not with the original efforts to develop the software, but with the continued evolution and maintenance of that software throughout its lifetime.



Good Programming Practice 1.1

Write your programs in a simple and straightforward manner. This is sometimes referred to as KIS (“keep it simple”). One key aspect of keeping it simple is another interpretation of KIS—“keep it small.” Do not “stretch” the language by trying bizarre us-es.



Good Programming Practice 1.2

Read the documentation for the Visual Basic version you are using to access VB's rich collection of features.



Error-Prevention Tip 1.2

Your computer and Visual Basic compiler are good teachers. If you are not sure how a feature works, even after studying the documentation, experiment and see what happens. Study each error or warning message and adjust the code accordingly.



1.8 C, C++ and Java

- ***C***
 - developed by **Dennis Ritchie** at **Bell Laboratories**
 - development language of the **UNIX** operating system
 - virtually all new major operating systems are written in **C** and/or **C++**
- ***C++***
 - developed by **Bjarne Stroustrup** in early **1980s**
 - “spruce up” the **C** language and provides capabilities for **object-oriented programming**
- ***Java***
 - developed by **Sun Microsystems** in **1991**
 - Sun saw the immediate potential of using **Java** to add dynamic content (e.g., interactivity, animations and the like) to web pages
 - Sun formally announced **Java** at an industry conference in **May 1995**
 - **Java** is now used to
 - develop large-scale enterprise applications
 - enhance the functionality of web servers
 - provide applications for consumer devices



1.9 BASIC, Visual Basic, Visual C++, C# and .NET

- **BASIC**
 - **Developed in the mid-1960s at Dartmouth College**
 - **Primary purpose was to familiarize novices with programming techniques**
- **Microsoft's Visual Basic language**
 - **Based on Basic**
 - **Has become one of the most popular programming languages in the world**
- **Microsoft's .NET platform**
 - **Provides the capabilities developers need to create computer applications that can execute on computers distributed across the Internet**
 - **Visual Basic (based on the original BASIC)**
 - **Visual C++ (based on C++)**
 - **Visual C# (based on C++ and Java)**



1.10 Software Technologies

- **Agile Software Development**
 - Set of methodologies that try to get software implemented quickly
 - Agile Alliance (www.agilealliance.org)
 - Agile Manifesto (www.agilemanifesto.org)
- **Refactoring**
 - Reworking code to make it clearer and easier to maintain while preserving its
- **Design patterns**
 - Proven architectures for constructing flexible and maintainable object-oriented software
 - Open source code



1.10 Software Technologies (Cont.)

- **Linux**
 - Open source operating system
- **Apache**
 - Most popular open source web server
- **MySQL**
 - Open source database management system
- **PHP**
 - Most popular open source server-side “scripting” language for developing Internet-based applications
- **LAMP**
 - Linux, Apache, MySQL and PHP (or Perl or Python)
- **Game programming**
 - Software techniques used in game programming Adobe Flash CS3
- **Ruby on Rails**
 - Combines the scripting language Ruby with the Rails web application framework
 - Developed by 37Signals
- **Software as a Service (SaaS)**
 - Software runs on servers elsewhere on the Internet
 - Salesforce.com, Google, Microsoft and 37Signals all offer SaaS

