Introduction to Computers, the Internet and Visual Basic
The chief merit of language is clearness.
– Galen

High thoughts must have high language.
– Aristophanes

Our life is frittered away with detail. . . . Simplify, simplify.
– Henry David Thoreau
My object all sublime
I shall achieve in time.
– W. S. Gilbert

Man is still the most extraordinary computer of all.
– John F. Kennedy
In this chapter you will learn:

- Basic hardware and software concepts.
- The different types of programming languages.
- Which programming languages are most widely used.
- The history of the Visual Basic programming language.
- Some basics of object technology.
OBJECTIVES

- The history of the UML—the industry-standard object-oriented system modeling language.
- The history of the Internet and the World Wide Web.
- The motivation behind and an overview of Microsoft’s .NET initiative, which involves the Internet in developing and using software systems.
- To test-drive a Visual Basic 2008 application that enables you to draw on the screen.
1.1 Introduction

• Computer use is increasing in almost every field.
• Computing costs have been decreasing due to rapid developments in both hardware and software technologies.
• Silicon chip technology has made computing economical for worldwide personal use.
1.2 What Is a Computer?

- A computer performs calculations and makes logical decisions.
- Computers process data using sets of instructions called computer programs.
  - These programs are specified by people known as computer programmers.
- A computer consists of various devices referred to as hardware.
- Programs that run on a computer are software.
1.3 Computer Organization

• The **input unit** obtains information from **input devices**, such as the keyboard and the mouse.

• The **output unit** takes information and places it on **output devices**, making the information available to the user.

• The **memory unit** stores data while an application is running.
  
  – **Random-access memory (RAM)** is primary memory.
  
  – Primary memory is usually **volatile**, and is erased when the machine is powered off.
1.3 Computer Organization (Cont.)

- The **central processing unit (CPU)** supervises the operation of the other sections.
- The **arithmetic and logic unit (ALU)** performs calculations.
  - It also makes decisions, such as determining whether two items stored in memory are equal.
- The **secondary storage unit** is the long-term storage of the computer in devices such as hard drives and DVD drives.
1.4 Early Operating Systems

• Computers of the 1950s could perform only one job or task at a time.
• Users submitted their jobs on decks of punched cards.
• Operating systems make using computers more convenient. Early operating systems increased the throughput computers could process.
• Many jobs or tasks could share the resources of a computer through multiprogramming—the simultaneous operation of many jobs.
• Timesharing allowed multiple users to access a computer through separate terminals.
  – The computer runs a small portion of one user’s job, then moves on to service the next user.
1.5 Personal Computing, Distributed Computing and Client/Server Computing

• Silicon chip technology made it possible for computers to be more economical.

• Apple Computer and IBM introduced rival personal computers. IBM introduced businesses to the benefits of personal computing.

• Computers communicated over telephone lines and local area networks (LANs).

• Distributed computing allowed an organization’s computing to be distributed over a network.

• Today, computers called servers offer a common data store used by client computers.
1.6 Hardware Trends

• **Moore’s Law** is the ongoing trend for computers to get more advanced while prices fall.

• Moore’s Law is especially true in relation to the amount of memory that computers have for programs, the amount of secondary storage they have, and their processor speeds.

• Microprocessor chip technology laid the groundwork for significant improvements in productivity.
1.7 Microsoft’s Windows Operating System

• In 1981, Microsoft released the first version of its DOS operating system.

• In the mid-1980s, Microsoft developed the Windows operating system, a graphical user interface built on top of DOS.

• The Windows operating system became especially popular after the 1993 release of Windows 3.1.

• Windows virtually cornered the operating systems market by the late 1990s.
1.7 Microsoft’s Windows Operating System (Cont.)

• Windows XP combined Microsoft’s corporate and consumer operating system lines.

• Windows Vista is Microsoft’s latest operating system offering.

• The biggest competitor to the Windows operating system is Linux.
  – Linux is a free, open-source operating system.
  – The source code for Linux is freely available to users, and they can modify it to fit their needs.
1.8 Machine Languages, Assembly Languages and High-Level Languages

• Programmers write instructions in various programming languages.

• Computer languages can be divided into three general types:
  – Machine languages
  – Assembly languages
  – High-level languages
1.8 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- A computer can directly understand only its own machine language.
- Machine languages generally consist of streams of numbers (ultimately reduced to binary 1s and 0s).
- Machine-language programs are nearly incomprehensible to humans:
  
  \[
  +1300042774 \\
  +1400593419 \\
  +1200274027
  \]
1.8 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- Machine-language programming proved to be slow and error prone.
- English-like abbreviations form the basis of assembly languages.
- Assemblers convert assembly-language programs to machine language.

```
LOAD    BASEPAY
ADD     OVERPAY
STORE   GROSSPAY
```

- Computers cannot understand assembly-language code until it is translated into machine language.
1.8 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- To speed up the programming process, **high-level languages** were developed.
- **Compilers** convert high-level-language programs into machine language.
- High-level languages look almost like everyday English:
  \[ \text{grossPay} = \text{basePay} + \text{overTimePay} \]
- Visual Basic is one of the world’s most popular high-level programming languages.
### 1.8 Machine Languages, Assembly Languages and High-Level Languages (Cont.)

<table>
<thead>
<tr>
<th>Language</th>
<th>Sample code/Instructions</th>
<th>Translator</th>
<th>From the programmer's perspective</th>
<th>From the computer’s perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine language</td>
<td>+1300042774 +1400593419 +1200274027</td>
<td>None</td>
<td>Slow, tedious, error prone</td>
<td>Natural language of a computer; the only language the computer can understand directly</td>
</tr>
<tr>
<td>Assembly language</td>
<td>LOAD ADD BASEPAY OVERPAY STORE GROSSPAY</td>
<td>Assembler</td>
<td>English-like abbreviations, easier to understand</td>
<td>Assemblers convert assembly language into machine language so the computer can understand</td>
</tr>
<tr>
<td>High-level language</td>
<td>grossPay = basepay + overTimePay</td>
<td>Compiler</td>
<td>Instructions resemble everyday English; single statements accomplish substantial tasks</td>
<td>Compiles convert high-level languages into machine language so the computer can understand</td>
</tr>
</tbody>
</table>

**Fig. 1.1** Comparing machine, assembly and high-level languages.
1.9 Visual Basic

- **BASIC** was developed in the mid-1960s for writing programs quickly and easily.
  - BASIC’s primary purpose was to teach programming to novices.
- When Microsoft developed the Windows® graphical user interface (GUI), the natural evolution of BASIC was to **Visual Basic**.
- Visual Basic programs are created with the use of a software tool called an **Integrated Development Environment (IDE)**.
1.9 Visual Basic (Cont.)

• The latest versions of Visual Basic are fully object oriented.

• Visual Basic is also event driven—it responds to user-initiated events such as mouse clicks, keystrokes and timers.

• It is a visual programming language—you’ll drag and drop objects like buttons and textboxes into place.
• **C** first gained widespread recognition as the development language of the UNIX operating system.
  – C is a hardware-independent language (i.e. it is portable to most computers).

• **C++** took the C language and provided capabilities for **object-oriented programming (OOP)**.

• **Objects** are reusable software **components** that model items in the real world.
  – Object-oriented programs are often easier to understand, correct and modify.
1.10  C, C++, Java and Visual C# (Cont.)

- Sun Microsystems began development of the **Java** programming language in 1991.
- Sun saw the possibility of using Java to add **dynamic content** to web pages.
- Java is now used to develop large-scale enterprise applications, to enhance the functionality of web pages and for many other purposes.
1.10 C, C++, Java and Visual C# (Cont.)

• Microsoft released the C# programming language for the .NET platform in 2000.

• C# has roots in C, C++ and Java, adapting the best features of each.

• C# is object oriented and has access to a powerful class library of prebuilt components.
1.11 Other High-Level Languages

- **Fortran** (Formula Translator) was developed in the mid-1950s to create scientific and engineering applications.

- **COBOL** is used primarily for business applications that require the manipulation of large amounts of data.
1.12 Structured Programming

• **Structured programming** is a disciplined approach to creating clear, correct and easy-to-modify programs.

• The **Pascal** programming language was designed for teaching structured programming.

• The **Ada** structured programming language was developed under the sponsorship of the U.S. Department of Defense (DOD).
1.13 Key Software Trend: Object Technology

- **Object technology** is a packaging scheme for creating meaningful software units.
- Almost any noun can be reasonably represented as a software object.
- Objects
  - have *properties* (also called *attributes*)
  - perform *actions* (also called *behaviors* or *methods*)
1.13 Key Software Trend: Object Technology (Cont.)

- **Classes** are types of related objects.
  - A class specifies the general format of its objects, and the properties and actions available to an object.
  - An object is related to its class in much the same way as a building is related to its blueprint.

- **Procedural programming languages** focus on actions rather than things.
1.13 Key Software Trend: Object Technology (Cont.)

• Properly designed classes can be reused on future projects.

• Using libraries of classes reduces the amount of effort required to implement new systems.

• Instead of worrying about minute details, you can focus on the behaviors and interactions of objects.
1.14 The Internet and the World Wide Web

• In the late 1960s, ARPA (Advanced Research Projects Agency of the Department of Defense) connected the main computer systems of universities and research institutions.

• This became known as the ARPAnet, the grandparent of today’s Internet.

• Its quick and easy communication came to be known as electronic mail (e-mail).
1.14 The Internet and the World Wide Web (Cont.)

• The protocol for communicating over the ARPAnet became known as the Transmission Control Protocol (TCP).

• TCP ensured that message “packets” were properly routed from sender to receiver.

• One challenge was to enable different networks to communicate with each other.
  – ARPA accomplished this by developing the Internet Protocol (IP).

• The combined set of protocols is now called TCP/IP.
1.14 The Internet and the World Wide Web (Cont.)

- Companies started to develop and enhance their Internet presence.
- This generated fierce competition among communications carriers and hardware and software suppliers.
- As a result, bandwidth has increased tremendously, while communication costs have plummeted.
1.14 The Internet and the World Wide Web (Cont.)

• The **World Wide Web** is a collection of hardware and software associated with the Internet.

• Tim Berners-Lee of CERN developed a technology for sharing information via the **HyperText Markup Language (HTML)**.

• He also wrote **HyperText Transfer Protocol (HTTP)**, the main internet communications protocol.
1.14 The Internet and the World Wide Web (Cont.)

• In 1994, Berners-Lee founded the **World Wide Web Consortium** (W3C), devoted to developing technologies for the World Wide Web.

• The W3C aims to make the web universally accessible regardless of disabilities, language or culture.
1.15 Extensible Markup Language (XML)

• HTML’s lack of extensibility frustrated developers. As a result, XML was developed by the W3C.

• Data independence is the essential characteristic of XML.
  – Because XML documents describe data, any application conceivably can process them.
  – Software developers are integrating XML into their applications to improve web functionality and interoperability.
  – XML is also being employed in databases.
• Applications employing XML can communicate with one another, if they can understand common XML vocabularies.

• The **Simple Object Access Protocol (SOAP)** is a technology for the transmission of objects.

• Since SOAP’s foundations are in XML and **HTTP**, it is supported on most computer systems.
1.16 Introduction to Microsoft .NET

- Microsoft’s .NET initiative uses the Internet and the web in the development, engineering, distribution and use of software.

- Applications in any .NET-compatible language can interact with each other.

- Microsoft’s ASP.NET technology allows you to create web applications.

- The .NET strategy allows programmers to concentrate on their specialties without having to implement every component of every application.
1.17 The .NET Framework and the Common Language Runtime

- The Microsoft **.NET Framework**:
  - manages and executes applications and web services
  - contains a class library (called the .NET Framework Class Library)
  - provides security and other programming capabilities.

- The **Common Language Runtime (CLR)**:
  - Programs are compiled first into **Microsoft Intermediate Language (MSIL)**.
  - When the application executes, the **just-in-time compiler** translates the MSIL in the executable file into machine-language code.
1.18 Test-Driving the Visual Basic Advanced Painter Application

- Open Windows Explorer and navigate to the C:\Examples\ch01 directory (Fig. 1.2).
- Double click the file name AdvancedPainter.exe to run the application.
1.18 Test-Driving the Visual Basic Advanced Painter Application (Cont.)

- A user interface uses GUI controls (Fig. 1.3).

**Fig. 1.3** | Visual Basic Advanced Painter application.
1.18 Test-Driving the Visual Basic Advanced Painter Application (Cont.)

- Click the Radio buttons labeled **Red** and **Small** to change the color and size of the brush.
- Press and hold down the left mouse button to draw, as shown in Fig. 1.4.

![Advanced Painter](image)

**Fig. 1.4** | Drawing with a new brush color.
• Click the Radio buttons labeled **Green** and **Large** to change the color and size of the brush (Fig. 1.5).

**Fig. 1.5** | Drawing with a new brush size.
1.18 Test-Driving the Visual Basic Advanced Painter Application (Cont.)

- Click the **RadioButton**s labeled **Blue** and **Medium**.
- Draw raindrops to complete the picture (Fig. 1.6).
- Close the application by clicking its **close box**.

**Fig. 1.6** | Finishing the drawing.
1.18 Test-Driving the Visual Basic Advanced Painter Application (Cont.)

• Figure 1.7 lists a few of the applications available in the examples and exercises.

• You’re encouraged to test-drive these applications.

<table>
<thead>
<tr>
<th>Application name</th>
<th>File to execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking Fees</td>
<td>Garage.exe</td>
</tr>
<tr>
<td>Tic Tac Toe</td>
<td>TicTacToe.exe</td>
</tr>
<tr>
<td>Drawing Stars</td>
<td>DrawStars.exe</td>
</tr>
<tr>
<td>Drawing Shapes</td>
<td>DrawShapes.exe</td>
</tr>
<tr>
<td>Drawing Polygons</td>
<td>DrawPolygons.exe</td>
</tr>
</tbody>
</table>

Fig. 1.7 | Examples of Visual Basic applications found in this book.
1.19 Introduction to Object Technology and the UML

- **Object-oriented design (OOD)** models software in terms similar to those that people use to describe real-world objects.

- OOD takes advantage of **inheritance** relationships:
  - A “convertible” object has the characteristics of an “automobile.”
  - The “convertible” also has unique properties.
1.19 Introduction to Object Technology and the UML (Cont.)

• OOD encapsulates attributes and operations into objects.

• An object’s attributes and operations are intimately tied together.

• Objects have the property of information hiding.
  – Objects communicate with one another across well-defined interfaces.
  – They do not usually access each other’s information directly.
• Each class contains data as well as the set of methods that manipulate that data and provide services to clients.

• The data components of a class are called attributes or fields. For example, a bank account class has an account number and a balance.

• A class is a “blueprint” for building objects of the class.

Software Engineering Observation 1.1

Reuse of existing classes when building new classes and programs saves time, money and effort. Reuse also helps programmers build more reliable and effective systems, because existing classes and components often have gone through extensive testing, debugging and performance tuning.
Introduction to Object-Oriented Analysis and Design (OOAD)

• Follow a process for analyzing your project’s requirements and developing a design that satisfies them.
• If this process involves an object-oriented point of view, it is called object-oriented analysis and design (OOAD).
• A group should agree on a strictly defined process for solving its problem and communicating its results.
What is the UML?

- The Unified Modeling Language (UML) models object-oriented systems graphically.
- The UML is extensible (i.e., capable of being enhanced with new features) and can be used with any OOAD process.
- UML allows all developers to express their designs with one standard set of notations.