foundations of
computer science

Behrouz Forouzan and Firouz Mosharraf

Visit the website at: www.thomsonlearning.co.uk/forouzan

second edition

COURSE TECHNOLOGY
CENGAGE Learning™

1 Introduction

The phrase *computer science* has a very broad meaning today. However, in this book, we define the phrase as "issues related to the computer". This introductory chapter first tries to find out what a computer is, then investigates other issues directly related to computers. We look first at the **Turing model** as a mathematical and philosophical definition of computation. We then show how today's computers are based on the **von Neumann model**. The chapter ends with a brief history of this culture-changing device… the computer.

Objectives

After studying this chapter, the student should be able to:

❑ Define the Turing model of a computer.

❑ Define the von Neumann model of a computer.

❑ Describe the three components of a computer: hardware, data, and software.

❑ List topics related to computer hardware.

❑ List topics related to data.

❑ List topics related to software.

❑ Discuss some social and ethical issues related to the use of computers.

❑ Give a short history of computers.

1.1 TURING MODEL

The idea of a universal computational device was first described by Alan Turing in 1937. He proposed that all computation could be performed by a special kind of a machine, now called a *Turing machine*. Although Turing presented a mathematical description of such a machine, he was more interested in the philosophical definition of computation than in building the actual machine. He based the model on the actions that people perform when involved in computation. He abstracted these actions into a model for a computational machine that has really changed the world.

Data processors

Before discussing the Turing model, let us define a computer as a **data processor**. Using this definition, a computer acts as a black box that accepts input data, processes the data, and creates output data (Figure 1.1). Although this model can define the functionality of a computer today, it is too general. In this model, a pocket calculator is also a computer (which it is, in a literal sense).

Input data ⟶ Computer ⟶ Output data

Figure 1.1   A single purpose computing machine

Another problem with this model is that it does not specify the type of processing, or whether more than one type of processing is possible. In other words, it is not clear how many types or sets of operations a machine based on this model can perform. Is it a specific-purpose machine or a general-purpose machine?

This model could represent a specific-purpose computer (or processor) that is designed to do a single job, such as controlling the temperature of a building or controlling the fuel usage in a car. However, computers, as the term is used today, are *general-purpose* machines. They can do many different types of tasks. This implies that we need to change this model into the Turing model to be able to reflect the actual computers of today.

Programmable data processors

The Turing model is a better model for a general-purpose computer. This model adds an extra element to the specific computing machine: the *program*. A **program** is a set of instructions that tells the computer what to do with data. Figure 1.2 shows the Turing model.

In the Turing model, the **output data** depends on the combination of two factors: the **input data** and the program. With the same input data, we can generate different outputs if we change the program. Similarly, with the same program, we can generate different outputs if we change the input data. Finally, if the input data
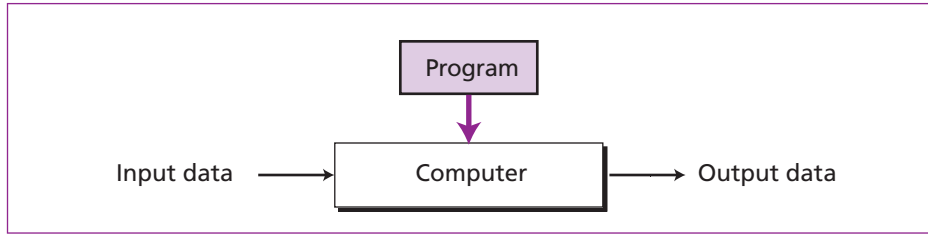
Figure 1.2 A computer based on the Turing model: programmable data processor

and the program remain the same, the output should be the same. Let us look at three cases.

Same program, different input data

Figure 1.3 shows the same sorting program with different input data. Although the program is the same, the outputs are different, because different input data is processed.
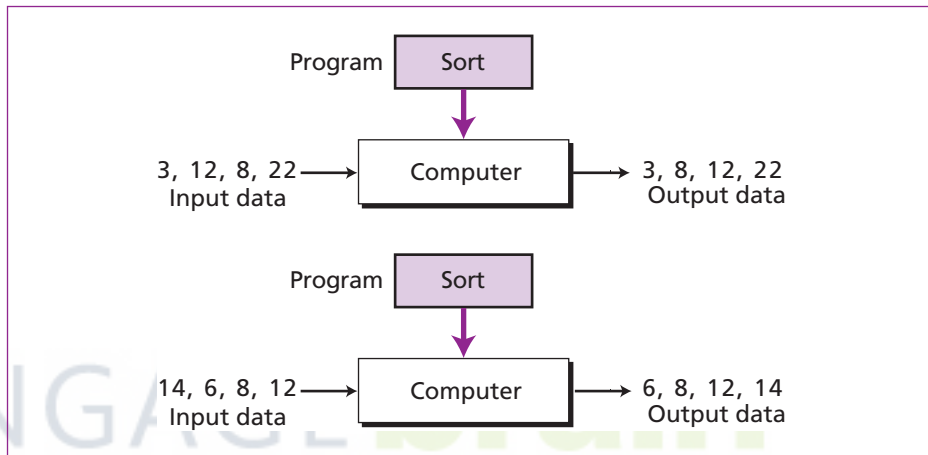
Figure 1.3 The same program, different data

Same input data, different programs

Figure 1.4 shows the same input data with different programs. Each program makes the computer perform different operations on the input data. The first program sorts the data, the second adds the data, and the third finds the smallest number.
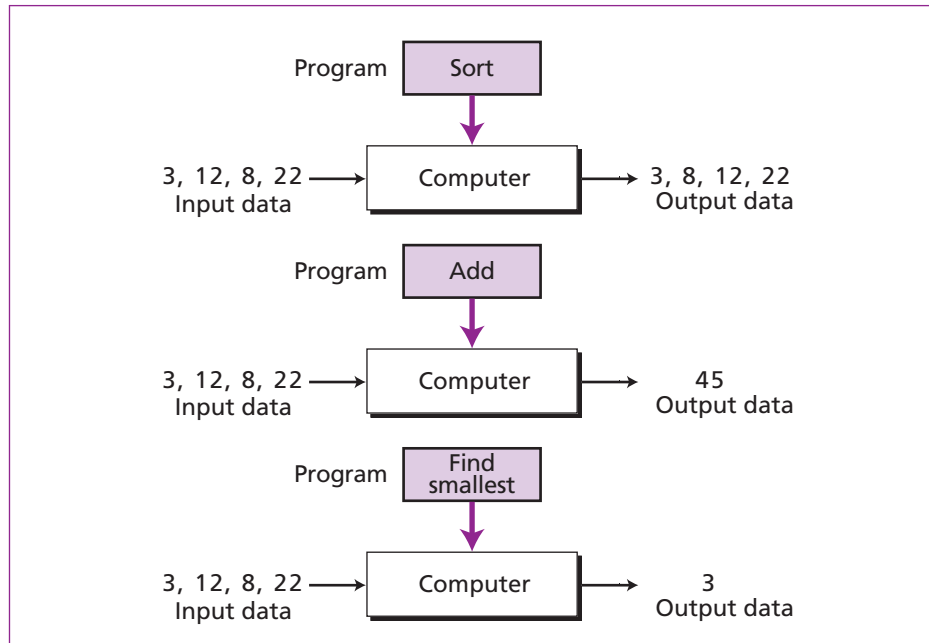
Figure 1.4   The same data, different programs

Same input data, same program

We expect the same result each time if both input data and the program are the same, of course. In other words, when the same program is run with the same input data, we expect the same output.

The universal Turing machine

A *universal Turing machine*, a machine that can do any computation if the appropriate program is provided, was the first description of a modern computer. It can be proved that a very powerful computer and a universal Turing machine can compute the same thing. We need only provide the data and the program—the description of how to do the computation—to either machine. In fact, a universal Turing machine is capable of computing anything that is computable.

1.2   VON NEUMANN MODEL

Computers built on the Turing universal machine store data in their memory. Around 1944–1945, John von Neumann proposed that, since program and data are logically the same, programs should also be stored in the memory of a computer.

Four subsystems

Computers built on the **von Neumann model** divide the computer hardware into four subsystems: memory, arithmetic logic unit, control unit, and input/output (Figure 1.5).
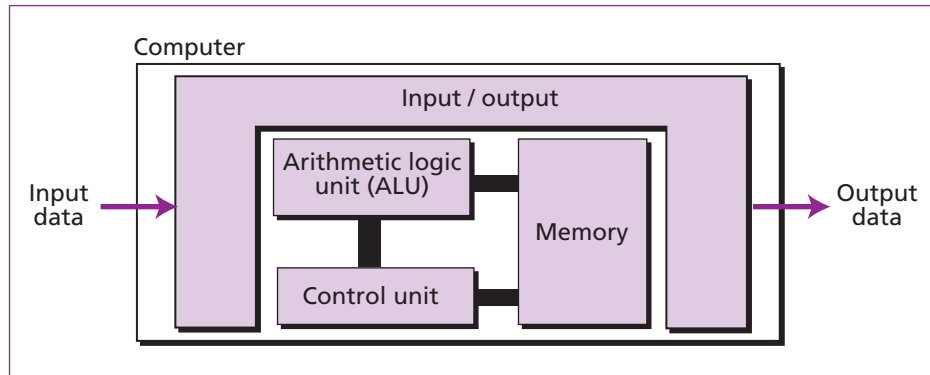
Figure 1.5   The von Neumann model

Memory

Memory is the storage area. This is where programs and data are stored during processing. We discuss the reasons for storing programs and data later in the chapter.

Arithmetic logic unit

The **arithmetic logic unit (ALU)** is where calculation and logical operations take place. For a computer to act as a data processor, it must be able to do arithmetic operations on data (such as adding a list of numbers). It should also be able to do logical operations on data, as we will see in Chapter 4.

Control unit

The **control unit** controls the operations of the memory, ALU, and the input/output subsystem.

Input / output

The input subsystem accepts input data and the program from outside the computer, while the output subsystem sends the results of processing to the outside world. The definition of the input/output subsystem is very broad: it also includes secondary storage devices such as disk or tape that store data and programs for processing. When a disk stores data that results from processing, it is considered an output device: when data is read from the disk, it is considered an input device.

The stored program concept

The von Neumann model states that the program must be stored in memory. This is totally different from the architecture of early computers in which only the data was stored in memory: the programs for their tasks were implemented by manipulating a set of switches or by changing the wiring system.

The memory of modern computers hosts both a program and its corresponding data. This implies that both the data and programs should have the same format, because they are stored in memory. In fact, they are stored as *binary* patterns in memory—a sequence of 0s and 1s.

Sequential execution of instructions

A program in the von Neumann model is made of a finite number of **instructions**. In this model, the control unit fetches one instruction from memory, decodes it,

then executes it. In other words, the instructions are executed one after another. Of course, one instruction may request the control unit to jump to some previous or following instruction, but this does not mean that the instructions are not executed sequentially. Sequential execution of a program was the initial requirement of a computer based on the von Neumann model. Today's computers execute programs in the order that is most efficient.

1.3 COMPUTER COMPONENTS

We can think of a computer as being made up of three components: computer hardware, data, and computer software.

Computer hardware

Computer hardware today has four components under the von Neumann model, although we can have different types of memory, different types of input/output subsystems, and so on. We discuss computer hardware in more detail in Chapter 5.

Data

The von Neumann model clearly defines a computer as a data processing machine that accepts the input data, processes it, and outputs the result.

Storing data

The von Neumann model does not define how data must be stored in a computer. If a computer is an electronic device, the best way to store data is in the form of an electrical signal, specifically its presence or absence. This implies that a computer can store data in one of two states.

Obviously, the data we use in daily life is not just in one of two states. For example, our numbering system uses digits that can take one of ten states (0 to 9). We cannot (as yet) store this type of information in a computer: it needs to be changed to another system that uses only two states (0 and 1). We also need to be able to process other types of data (text, image, audio, video). These also cannot be stored in a computer directly, but need to be changed to the appropriate form (0s and 1s).

In Chapters 3, we will learn how to store different types of data as a binary pattern, a sequence of 0s and 1s. In Chapter 4, we show how data is manipulated, as a binary pattern, inside a computer.

Organizing data

Although data should be stored in only one form inside a computer, a binary pattern, data outside a computer can take many forms. In addition, computers (and the notion of data processing) have created a new field of study known as *data organization*, which asks the question: can we organize our data into different entities and formats before storing it inside a computer? Today, data is not treated as a flat sequence of information. Instead, data is organized into small units, small units are organized into larger units, and so on. We will look at data from this point of view in Chapters 11–14.

Computer software

The main feature of the Turing or von Neumann models is the concept of the *program*. Although early computers did not store the program in the computer's memory, they did use the concept of programs. *Programming* those early computers meant changing the wiring systems or turning a set of switches on or off. Programming was therefore a task done by an operator or engineer before the actual data processing began.

Programs must be stored

In the von Neumann model programs are stored in the computer's memory. Not only do we need memory to hold data, but we also need memory to hold the program (Figure 1.6).
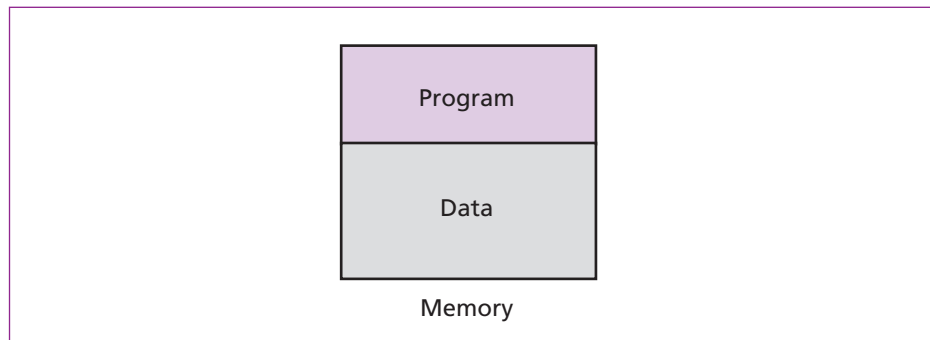
Figure 1.6   Program and data in memory

A sequence of instructions

Another requirement of the model is that the program must consist of a sequence of instructions. Each instruction operates on one or more data items. Thus, an instruction can change the effect of a previous instruction. For example, Figure 1.7 shows a program that inputs two numbers, adds them, and prints the result. This program consists of four individual instructions.

1. Input the first number into memory.

2. Input the second number into memory.

3. Add the two together and store the result in memory.

4. Output the result.

Program

Figure 1.7   A program made of instructions

We might ask why a program must be composed of instructions. The answer is reusability. Today, computers do millions of tasks. If the program for each task was

an independent entity without anything in common with other programs, programming would be difficult. The Turing and von Neumann models make programming easier by defining the different instructions that can be used by computers. A programmer can then combine these instructions to make any number of programs. Each program can be a different combination of different instructions.

Algorithms

The requirement for a program to consist of a sequence of instructions made programming possible, but it brought another dimension to using a computer. A programmer must not only learn the task performed by each instruction, but also learn how to combine these instructions to do a particular task. Looking at this issue differently, a programmer must first solve the problem in a step-by-step manner, then try to find the appropriate instruction (or series of instructions) to implement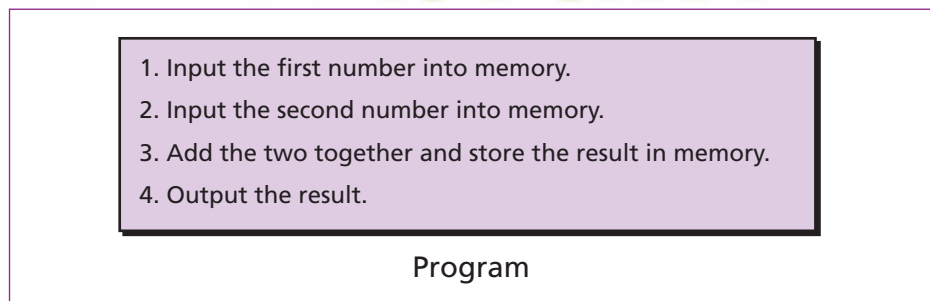 those steps. This step-by-step solution is called an **algorithm**. Algorithms play a very important role in computer science and are discussed in Chapter 8.

Languages

At the beginning of the computer age there was only one computer language, *machine language*. Programmers wrote instructions (using binary patterns) to solve a problem. However, as programs became larger, writing long programs using these patterns became tedious. Computer scientists came up with the idea of using symbols to represent binary patterns, just as people use symbols (words) for commands in daily life. Of course, the symbols used in daily life are different than those used in computers. So the concept of **computer languages** was born. A natural language such as English is rich and has many rules to combine words correctly: a computer language, on the other hand, has a more limited number of symbols and also a limited number of words. We will study computer languages in Chapter 9.

Software engineering

Something that was not defined in the von Neumann model is **software engineering**, which is the design and writing of *structured programs*. Today it is not acceptable just to write a program that does a task: the program must follow strict rules and principles. We discuss these principles, collectively known as *software engineering*, in Chapter 10.

Operating systems

During the evolution of computers, scientists became aware that there was a series of instructions common to all programs. For example, instructions to tell a computer where to receive data and where to send data are needed by almost all programs. It is more efficient to write these instructions only once for the use of all programs. Thus the concept of the **operating system** emerged. An operating system originally worked as a manager to facilitate access to the computer's components by a program, although today operating systems do much more. We will learn about them in Chapter 7.

1.4 HISTORY

In this section we briefly review the history of computing and computers. We divide this history into three periods.

Mechanical machines (before 1930)

During this period, several computing machines were invented that bear little resemblance to the modern concept of a computer.

❏ In the 17th century, Blaise Pascal, a French mathematician and philosopher, invented Pascaline, a mechanical calculator for addition and subtraction operations. In the 20th century, when Niklaus Wirth invented a structured programming language, he called it Pascal to honor the inventor of the first mechanical calculator.

❏ In the late 17th century, German mathematician Gottfried Leibnitz invented a more sophisticated mechanical calculator that could do multiplication and division as well as addition and subtraction. It was called Leibnitz' Wheel.

❏ The first machine that used the idea of storage and programming was the Jacquard loom, invented by Joseph-Marie Jacquard at the beginning of the 19th century. The loom used punched cards (like a stored program) to control the raising of the warp threads in the manufacture of textiles.

❏ In 1823, Charles Babbage invented the Difference Engine, which could do more than simple arithmetic operations—it could solve polynomial equations too. Later, he invented a machine called the Analytical Engine that, to some extent, parallels the idea of modern computers. It had four components: a mill (corresponding to a modern ALU), a store (memory), an operator (control unit), and output (input/output).

❏ In 1890, Herman Hollerith, working at the US Census Bureau, designed and built a programmable machine that could automatically read, tally, and sort data stored on punched cards.

The birth of electronic computers (1930–1950)

Between 1930 and 1950, several computers were invented by scientists who could be considered the pioneers of the electronic computer industry.

Early electronic computers

The early computers of this period did not store the program in memory—all were programmed externally. Five computers were prominent during these years:

❏ The first special-purpose computer that encoded information electrically was invented by John V. Atanasoff and his assistant Clifford Berry in 1939. It was called the ABC (Atanasoff Berry Computer) and was specifically designed to solve a system of linear equations.

❏ At the same time, a German mathematician called Konrad Zuse designed a general-purpose machine called Z1.

❏ In the 1930s, the US Navy and IBM sponsored a project at Harvard University under the direction of Howard Aiken to build a huge computer called Mark I. This computer used both electrical and mechanical components.

❏ In England, Alan Turing invented a computer called Colossus that was designed to break the German Enigma code.

❑   The first general-purpose, totally electronic computer was made by John Mauchly and J. Presper Eckert and was called ENIAC (Electronic Numerical Integrator and Calculator). It was completed in 1946. It used 18,000 vacuum tubes, was 100 feet long by 10 feet high, and weighed 30 tons.

Computers based on the von Neumann model

The preceding five computers used memory only for storing data, and were programmed externally using wires or switches. John von Neumann proposed that the program and the data should be stored in memory. That way, every time we use a computer to do a new task, we need only change the program instead of rewiring the machine or turning hundreds of switches on and off.

The first computer based on von Neumann's ideas was made in 1950 at the University of Pennsylvania and was called EDVAC. At the same time, a similar computer called EDSAC was built by Maurice Wilkes at Cambridge University in England.

Computer generations (1950–present)

Computers built after 1950 more or less follow the von Neumann model. They have become faster, smaller, and cheaper, but the principle is almost the same. Historians divide this period into generations, with each generation witnessing some major change in hardware or software (but not in the model).

First generation

The first generation (roughly 1950–1959) is characterized by the emergence of commercial computers. During this time, computers were used only by professionals. They were locked in rooms with access limited only to the operator or computer specialist. Computers were bulky and used vacuum tubes as electronic switches. At this time, computers were affordable only by big organizations.

Second generation

Second-generation computers (roughly 1959–1965) used transistors instead of vacuum tubes. This reduced the size of computers, as well as their cost, and made them affordable to small and medium-size corporations. Two high-level programming languages, FORTRAN and COBOL (see Chapter 9), were invented and made programming easier. These two languages separated the programming task from the computer operation task. A civil engineer, for example could write a FORTRAN program to solve a problem without being involved in the electronic details of computer architecture.

Third generation

The invention of the **integrated circuit** (transistors, wiring, and other components on a single chip) reduced the cost and size of computers even further. *Minicomputers* appeared on the market. Canned programs, popularly known as *software packages*, became available. A small corporation could buy a package, for example for accounting, instead of writing its own program. A new industry, the software industry, was born. This generation lasted roughly from 1965 to 1975.

Fourth generation

The fourth generation (approximately 1975–1985) saw the appearance of *microcomputers*. The first desktop calculator, the Altair 8800, became available in 1975. Advances in the electronics industry allowed whole computer subsystems to fit on

a single circuit board. This generation also saw the emergence of computer net-works (see Chapter 6).

Fifth generation

This open-ended generation started in 1985. It has witnessed the appearance of laptop and palmtop computers, improvements in secondary storage media (CD-ROM, DVD and so on), the use of multimedia, and the phenomenon of virtual reality.

1.5 SOCIAL AND ETHICAL ISSUES

Computer science has created some peripheral issues, the most prevalent of which can be categorized as social and ethical issues.

Social issues

Computers have created some controversy. We introduce some of these arguments here.

Dependency

Computer science has definitely changed our society. Based on some surveys, more than half of the households in the US use a computer to access the Internet. Does this mean that a society in which the majority of people have access to a computer is considered a *better* society? Some people think that the use of computers is inevitable in everyone's life: life is more difficult without it, while others think that computers have created a kind of dependency. The latter group think that this new dependency, like others, makes people's lives more difficult.

Social justice

Social justice is another issue we often hear about. The advocates of this issue argue that using computers at home is a luxury that not all people can afford. The cost of a computer, peripheral devices, and a monthly charge for Internet access is an extra burden on low-income people. This means that computers have created a service only for middle or high-income people, while low-income people are deprived of it. The opponents of this idea argue that the same issue was applied when other modern communication services were created, such as telephone and television. This group argue that, with the advance of technology, one day everyone will be able to afford a computer and access to the Internet.

Digital divide

The concept of a **digital divide** covers both the issues of dependency and social justice discussed above. The concept divides society into two groups: those who are electronically connected to the rest of society and those who are not. Someone in the first group communicates with people in the group through e-mail and uses the Internet for business and entertainment. Someone in the second group uses "snail mail" and the telephone for communication, and enjoys sitting in a cinema to see a movie instead of downloading it from a cable network. This digital divide is gradually disappearing in industrial countries, but experts believe that it will stay for a long time in developing countries.

Ethical issues

Computers have created some ethical issues. We introduce some of these here.

Privacy

Computers allow communication between two parties to be done electronically. However, much needs to be done to make this type of communication private. Society is paying a high price for private electronic communication. Network security (Chapter 16) may create this type of privacy, but it needs effort and cost a lot.

Copyright

Another ethical issue in a computerized society is copyright: who owns data? The Internet has created opportunities to share ideas, but has also brought a further ethical issue: electronic copyright.

Computer crime

Like any innovation, computers and information technology have created new types of crime. Hackers have been able to access many computers in the world and have stolen a lot of money. Virus creators design new viruses to be sent through the Internet and damage the information stored in computers. Although there are many anti-virus programs in use today, society is paying a high price for this type of crime, which did not exist before the computer and Internet era.

1.6   COMPUTER SCIENCE AS A DISCIPLINE

With the invention of computers, a new discipline has evolved: *computer science*. Like any other discipline, computer science has now divided into several areas. We can divide these areas into two broad categories: *system areas* and *applications areas*. System areas cover those areas that are directly related to the creation of hardware and software, such as *computer architecture*, *computer networking*, *security issues*, *operating systems*, *algorithms*, *programming languages*, and *software engineering*. Applications areas cover those that are related to the *use* of computers, such as *databases* and *artificial intelligence*. This book is a breadth-first approach to all these areas. After reading the book, the reader should have enough information to select the desired area of specialty.

1.7   OUTLINE OF THE COURSE

After this introductory chapter, the book is divided into five parts.

Part I: Data representation and operation

This part includes Chapters 2, 3, and 4. Chapter 2 discusses number systems, how a quantity can be represented using symbols. Chapter 3 discusses how different data is stored inside the computer. Chapter 4 discusses some primitive operations on *bits*.

Part II: Computer hardware

This part includes Chapters 5 and 6. Chapter 5 gives a general idea of computer hardware, discussing different computer organizations. Chapter 6 shows how individual computers are connected to make computer networks, and *internetworks* (internets). In particular, this chapter explores some subjects related to the Internet and its applications.

Part III: Computer software

This part includes Chapters 7, 8, 9, and 10. Chapter 7 discusses operating systems, the system software that controls access to the hardware by users—either human or application programs. Chapter 8 shows how problem solving is reduced to writing an algorithm for the problem. Chapter 9 takes a journey through the list of contemporary programming languages. Finally, Chapter 10 is a review of software engineering, the engineering approach to the development of software.

Part IV: Data organization and abstraction

This part complements Part I. In computer science, *atomic* data is collected into records, files, and databases. Data *abstraction* allows the programmer to create abstract notions about data. Part IV includes Chapters 11, 12, 13, and 14. Chapter 11 discusses data structures, collecting data of the same or different types under one category. Chapter 12 discusses abstract data types. Chapter 13 shows how different file structures can be used for different purposes. Finally, Chapter 14 discusses databases.

Part V: Advanced topics

Part V gives an overview of advanced topics, topics that students of computer science will encounter later in their education. This part covers Chapters 15, 16, 17, and 18. Chapter 15 discusses data compression, which is prevalent in today's data communications. Chapter 16 explores some issues to do with security, which is becoming more and more important when we communicate over insecure channels. Chapter 17 discusses the theory of computation: what can and cannot be computed. Finally Chapter 18 gives some idea of artificial intelligence, a topic with day-to-day challenges in computer science.

1.8 RECOMMENDED READING

For more details about the subjects discussed in this chapter, the following books are recommended:

❏ Schneider G M and Gersting J L: *Invitation to Computer Science*, Boston, MA: Course Technology, 2004

❏ Dale N and Lewis J: *Computer Science Illuminated*, Sudbury, MA: Jones and Bartlett, 2004

❏ Patt Y and Patel S: *Introduction to Computing Systems*, New York: McGraw-Hill, 2004

1.9   KEY TERMS

This chapter has introduced the following key terms, which are listed here with the pages on which they first occur:

algorithm 8	arithmetic logic unit (ALU) 5
computer languages 8	control unit 5
data processor 2	digital divide 11
input data 2	instruction 5
integrated circuit 10	memory 5
operating system 8	output data 2
program 2	software engineering 8
Turing model 1	von Neumann model 5

1.10   SUMMARY

- The idea of a universal computational device was first given by Alan Turing in 1937. He proposed that all computation can be performed by a special kind of a machine, now called a Turing machine.

- The von Neumann model defines a computer as four subsystems: memory, arithmetic logic unit, control unit, and input/output. The von Neumann model states that the program must be stored in memory.

- We can think of a computer as made up of three components: computer hardware, data, and computer software.

- The history of computing and computers can be divided into three periods: the period of mechanical machines (before 1930), the period of electronic computers (1930–1950), and the period that includes the five modern computer generations.

- Computer science has created some peripheral issues, the most prevalent of which can be categorized as social and ethical issues.

- With the invention of computers a new discipline has evolved, *computer science*, which is now divided into several areas.

1.11   PRACTICE SET

Review questions

1. Define a computer based on the Turing model.
2. Define a computer based on the von Neumann model.
3. What is the role of a program in a computer that is based on the Turing model?
4. What is the role of a program in a computer based on the von Neumann model?
5. What are the various subsystems of a computer?
6. What is the function of the memory subsystem in a computer?
7. What is the function of the ALU subsystem in a computer?
8. What is the function of the control unit subsystem in a computer?
9. What is the function of the input/output subsystem in a computer?
10. Briefly describe the five generations of computers.

Multiple-choice questions

11. The _____ model is the basis for today's computers.
    a. Leibnitz
    b. von Neumann
    c. Pascal
    d. Charles Babbage

12. In a computer, the _____ subsystem stores data and programs.
    a. ALU
    b. input/output
    c. memory
    d. control unit

13. In a computer, the _____ subsystem performs calculations and logical operations.
    a. ALU
    b. input/output
    c. memory
    d. control unit

14. In a computer, the _____ subsystem accepts data and programs and sends processing results to output devices.
    a. ALU
    b. input/output
    c. memory
    d. control unit

15. In a computer, the _____ subsystem serves as a manager of the other subsystems.
    a. ALU
    b. input/output
    c. memory
    d. control unit

16. According to the von Neumann model, _____ stored in memory.
    a. only data is
    b. only programs are
    c. data and programs are
    d. (none of the above)

17. A step-by-step solution to a problem is called _____.
    a. hardware
    b. an operating system
    c. a computer language
    d. an algorithm

18. FORTRAN and COBOL are examples of _____.
    a. hardware
    b. operating systems
    c. computer languages
    d. algorithms

19. A 17th-century computing machine that could perform addition and subtraction was the _____.
    a. Pascaline
    b. Jacquard loom
    c. Analytical Engine
    d. Babbage machine

20. _____ is a set of instructions in a computer language that tells the computer what to do with data.
    a. An operating system
    b. An algorithm
    c. A data processor
    d. A program

21. _____ is the design and writing of a program in structured form.
    a. Software engineering
    b. Hardware engineering
    c. Algorithm development
    d. Instructional architecture

22. The first electronic special-purpose computer was called _____.
    a. Pascal
    b. Pascaline
    c. ABC
    d. ENIAC

23. One of the first computers based on the von Neumann model was called _____.
    a. Pascal
    b. Pascaline
    c. ABC
    d. EDVAC

24. The first computing machine to use the idea of storage and programming was called _____.
    a. the Madeline
    b. EDVAC
    c. the Babbage machine
    d. the Jacquard loom

25. _____ separated the programming task from computer operation tasks.
    a. Algorithms
    b. Data processors
    c. High-level programming languages
    d. Operating systems

Exercises

26. Explain why a computer cannot solve a problem for which there is no solution outside the computer.

27. If a small cheap computer can do the same thing as a large expensive computer, why do people need to have a large one.

28. Do some research and find out whether the Pascaline calculator is a computer according to the Turing model.

29. Do some research and find out whether Leibnitz's Wheel is a computer according to the Turing model.

30. Do some research and find of whether the Jacquard Loom is a computer according to the Turing model.

31. Do some research and find out whether Babbage's Analytical Engine is a computer according to the von Neumann model.

32. Do some research about the ABC computer and find out whether this computer followed the von Neumann model.

33. Do some research and find out in which computer generation keyboards originated.

34. Do some research and find out in which computer generation printers originated.

35. According to the von Neumann model, can the hard disk of today be used as input or output? Explain.

36. A programming language has ten different instructions. How many five-instruction programs can be written in this language if no instruction is repeated? How many seven-instruction programs?

37. Which is more valuable today to an organization: hardware, software, or data?