Advanced Business BASIC

For Microcomputers

Chao C. Chien

Advanced Business BASIC
For Microcomputers

Chao C. Chien
Los Angeles City College

ISBN 0-256-03175-4

Library of Congress Catalog Card No. 84–81736

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0 K 2 1 0 9 8 7 6 5

Preface

During the last several years, we have been privileged to witness one of the most remarkable events in computer history: the coming of age of the microcomputer. In less than a mere five-year span, the microcomputer, or what used to be popularly called the home computer, has matured from a little electronic game player, which also happened to possess some programming capabilities, into a powerful general-purpose computing machine, which equals some of the older and much larger computers in processing power, memory capacity, and peripheral equipment sophistication. Ten years ago a computing professional would show skepticism if he or she were told the entire computer would one day be replaced by a machine one third of its size. Now the whole CPU is etched onto a silicon chip the size of a fingernail, and a microcomputer is literally small enough to sit on one's lap and yet is capable of performing some of the most intricate tasks demanded of a certified accountant. In short, while the full-feature computer has dramatically shrunk in size, the microcomputer has grown into a serious business tool that industry giants such as IBM and Digital Equipment Corp. have had to reckon with.

To whom or what does the microcomputer owe its phenomenal success? Without a doubt, the pure and unabashed technological advancement and breakthroughs in the electronic field must be recognized for their contribution. But these breakthroughs were made possible only by the acquisition of knowledge over many years and by its documentation in textbooks in such an orderly and logical manner as to allow the achievements and failures of early generations of developers to be learned by the later generations.

The success of the microcomputer, as in the development of the earlier models, also owes much to the parallel development of another component of the computer besides the hardware: the programming languages, in particular, BASIC. The maturation of BASIC has been phenomenal. During the same time span as the hardware development, BASIC has managed to transform itself from containing two handfuls of keywords with limited finesse intended for the less serious programmers (BASIC stands for Beginner's All-purpose Symbolic Instruction Code) into a full-grown all-purpose language rivaling such stalwart languages as FORTRAN and COBOL.

The development of BASIC also parallels the development of the microcomputer hardware in another aspect: Both are very much the products of the marketplace. Although BASIC was born under the environment of the academe, its growth is largely due to the efforts of the people of industry. Unlike the

other well-known high-level languages, there is no formal standardization for BASIC. There are no such officially sanctioned and maintained versions as "BASIC 66" or "BASIC 77." No major conventions were held where eminent figures from the establishment sat down and decided what the code should be. BASIC was perfected by the shear force of free enterprise and competition. MBASIC, CBASIC, SBASIC, BASIC-Plus, Advanced BASIC, to name a few, are not institutional standards but, rather, brand names. In short, BASIC got better because the manufacturers of microcomputers needed better languages to run their powerful machines. The versions that are most versatile, logical, and practical therefore sold the most, and the most widely used become the de facto standard. The BASIC versions illustrated in this textbook are thus simply the versions most widely adopted by hardware manufacturers.

As the logical elements of a language increase and the language becomes more sophisticated, techniques in using them in combination are invented to produce powerful and specific effects. As standard applications programming techniques take hold, programming methods evolve. Besides knowing the fundamental commands and the well-established techniques and tricks, the programmer must also become well disciplined in the analytical thought processes and approaches in order to maximize the programming efficiency. The BASIC commands and the more elementary programming techniques and concepts, such as counting, looping, branching, and accumulating, are normally subjects of introductory programming courses. It is the programming approaches and the development of the analytical thought process that are the goals of this textbook. The thrust of this book is to introduce you to the logical and disciplined approach to looking at and understanding a problem, analyzing it, and finding a way to designing the solution programs. In fact three things will be emphasized: language features that are usually not included in an introductory course; advanced programming techniques; and the methods normally adopted in solving a business-oriented problem.

Although it is assumed that you already have a good foundation in elementary BASIC and a good understanding of the elementary business principles, the first part of the book is devoted to giving you an opportunity to refresh your memory on these subjects. Since the remainder and major part of this textbook is concerned with learning programming, which is essentially a skill, many work problems are designed for you to practice the newly learned material. The last part of the book concentrates on using the skills thus acquired to solve real business problems. It is hoped that when the course is finished, you will not only have acquired a valuable skill but also will be prepared to get a programming job and solve real-life problems.

Many thanks must be extended to Professors W. James Abbott, Jr., Broome Community College; Carolyn Hershberger, Olympic College; Robert W. Pew, Valencia Community College; William L. Bonney, Hudson Valley Community College; Laura Saret, Oakton Community College; Carol A. Rowey, Community College of Rhode Island; Anthony Basilico, Community College of Rhode Island; Robert B. Fetter, Yale University; Claude McMillan, University of Colorado; and Dr. Larry W. Arp, Indiana State University, Evansville, for their contributions to the selection, inclusion, organization, and emphasis of topics for this textbook.

Chao C. Chien

Contents

1 Introduction

After completing Chapter 1, you will be able to:

Describe the major MIS.

Distinguish between sequential and direct access file processing.

Billy's friends used to call him the "computer whiz" because he was the most knowledgeable about computers among the kids on the block. His college-age brother, however, called him the "computer junkie" because Billy spent practically all his extracurricular time playing video games or programming new games. Without a doubt Billy was bright, but Billy was also privileged. When he was 14 his father bought him a home computer equipped with a few alien-invader game cartridges and a small BASIC-language module. Billy was also especially lucky because he had never been afraid of the computer, as were some of the other kids. By the time he was 15 Billy had already written half a dozen new games in BASIC for his friends to play; and when he was 16 he had even taught himself quite a bit of programming in **assembly language** so he could make his monster heroes run faster.

When Billy graduated from high school, he already had about five years of programming "experience" behind him. With his programming savvy, Billy decided he was going to skip college and head straight for where the big bucks were. As a start, Billy persuaded his Uncle John to hire him in his wholesale garment firm as a summer replacement for the regular programmer; that way Billy could also build up some credentials for his resume. However, Billy was let go from the job after one week.

It is not known what really happened, but according to the other programmers in the department, Billy simply was not qualified for the position. He did not know what a **file** was or what was meant by a **two-dimensional** array. Tom, the manager of the department, wanted to give Billy a chance because Billy was the boss's nephew. But when Billy asked Tom whether he preferred to have the order entry-

menu more in the style of Pac-man or Donkey-Kong, Tom decided it was too much.

Billy is now in school working toward a degree in computer science, taking a first course in BASIC programming.

Qualification in programming is not necessarily gained from hours at the keyboard, although that certainly helps. What builds a programmer's ability is the knowledge for solving a practical problem. This includes knowing about the programming languages; the techniques for using the languages normally employed; the established methods used for handling various real-life problems, and, most important, the thinking process for designing the strategy to reach the solution. Programmers are good not because they can come up with tricks but because they get the job done in the most efficient manner; this they can do because they have the benefit of the experiences accumulated by previous programmers and a well-disciplined attitude toward problem solving.

An **applications programmer** is made by two basic factors: knowledge of the computer system and the proper approaches to solving a problem. Strictly and academically speaking, knowledge of the computer system is a requirement for the **systems analyst** and not the programmer. But for an applications programmer to be truly successful, knowing only the language codes is not enough for producing a program that solves a problem efficiently. An applications programmer must know how the computer system functions as a whole in order to design the optimal solution to a problem. By computer system, we do not simply mean the hardware. A computer system also includes the software and the people who use the hardware and software. In particular, the term *software* does not mean programs only. Software includes other conceptual items employed in data processing, such as the data that are needed to solve the problem—what they are, where they are, and how they are kept and arranged; the procedures for program usage and maintenance; and the available alternatives in case the program is rendered useless. Software must also relate to the particular equipment and personnel involved. For instance, if the data are stored on **tape**, then the program must not use commands that interact with a **disk**. If the program is to be used primarily by people untrained in data processing, then the program should be designed to be **user-friendly**.

In most introductory programming courses, the emphasis is on learning how to make the computer respond to the program. In introductory courses one learns the effects produced by the commands but seldom how to coordinate the commands in order to achieve specific application purposes, such as searching a data table or fitting a curve to a group of data points. In this course we will concentrate on the methods used to solve business (applications) problems using the BASIC language. In order to do so, we shall begin by understanding the meanings of **applications programming**.

1.1 WHAT IS AN APPLICATIONS PROGRAM?

Put simply, applications programming is programming for a real-life need. In more scientific terms, applications programming means solving a real-life problem or task by means of a computer, although a solution generated by other means may already exist. Applications programming thus implies using the computer as a tool for solving a problem that is not primarily computer-related. A game program, for example, is not an applications program because it does not purport to solve a real-life problem. An **operating system** is not an applications program because its purpose is to manage the internal activities of a computer; there is no noncomputer-related replacement for what it does. A program to help balance your checkbook, however, is an applications program because balancing the checkbook is a real-life activity, and the program is

merely a tool to help perform this task. A checkbook can be balanced without a computer. The computer merely makes the task a little easier to do.

Notice the words *problem* and *task*. These words suggest that such activities are either necessities in life or that the present noncomputer-oriented method of handling them is unsatisfactory in some way. Later in the course you will see that part of the concerns in writing applications programs is improvement over manual methods.

A business program is an applications program. In fact, it is a special kind of applications program. Specifically, a business program is a computer program whose purpose is to help solve a business-oriented problem. When implemented on a computer, such a program replaces people as the primary processor of data.

1.2 THE STANDARD DATA PROCESSING SYSTEM

Obviously a business engages in many activities, What is not obvious, however, is that smaller businesses do not necessarily engage in fewer activities than larger businesses; larger businesses, although larger in scope, are not always more complicated. To illustrate the point, let us take a look at two examples.

First, let us look at Contec, Incorporated, a primary contractor for the Defense Department with annual gross sales in the $2 billion range. What does Contec do? What are the more than 5,000 employees and the four IBM computers of Contec engaged in, day in and day out, 365 days a year?

Contec, Inc., is in the business of designing and manufacturing equipment for the United States Army. There are no salespeople to generate business. Contec does not need any. Instead, the company employs consulting engineers who have offices in Washington, D.C., and who work closely with the Defense Department. Contec's business comes in the form of contracts from the Defense Department. After obtaining the contracts, Contec proceeds to design and manufacture the parts needed by the army; this it does either in its own plants or by subcontracting some of the manufacturing out to the many satellite companies in its region, southern California. To say the least, Contec is large, and its operations are complicated.

Across town, two miles away from Contec, is Kindtharte Clinics Group, a nonprofit health care organization employing fewer than 45 people, doctors and nurses included. Kindtharte does not produce a product. On the contrary, it is a consumer of products. Every year it uses many pounds and cartons of drugs and medical supplies. Like Contec, Kindtharte has no sales force. Nor does it have customers; it has patients. Kindtharte gets no contract from the government. To survive, it has to rely heavily on contributions from the public. As a matter of fact, due to its nonprofit status, many people don't even consider Kindtharte a real business. Many times it has almost gone out of business.

Strangely perhaps, the definition of a business has nothing to do with making a profit. What the organization *does* defines a business. An organization is a business if it sells a product or a service for income, whether the income covers the cost or not. If the company loses money then it is at most a bad business, but a business nonetheless.

Both Contec and Kindtharte sell something. Contec sells tangible defense goods, and Kindtharte sells medical services. Both of them spend money to stay in business. Both own or rent their places of business. Both pay the phone company for phone services and the power and gas companies for utilities. To build tanks, its major product, Contec uses parts fabricated both in-house and supplied by subcontractors. Kindtharte, on the other hand, buys drug supplies from regional jobbers representing the major pharmaceutical firms. Contec buys raw materials from suppliers, while public-spirited citizens voluntarily donate to Kindtharte's blood bank. Have no doubt about it, though, profit or

nonprofit, both firms charge their customers or patients and pay salaries to their employees.

Contec, in truth, has only one customer; it has no need for an ordering desk. Contec does have, however, many lawyers on its payroll to scrutinize the contracts.

Kindtharte, on the other hand, has many patients. Most patients are long-term customers of Kindtharte because Kindtharte's services are paid for by employers who subscribe in group service form. Normally a small percentage of patients do not return for one reason or another, and sometimes new patients join the organization. Kindtharte takes payments from its patients in the form of employer contributions, personal checks, and, of course, cash. Kindtharte employs two full-time clerks just to take care of the patient records and the large amount of billing activity.

Contec is a manufacturing company. Kindtharte provides services. Contec employs more than 5,000 persons. Kindtharte employs only 45. They differ drastically in function and size, and, at the same time, their activities are radically different. Yet some activities are common to both of them, indeed, common to all business organizations: every business must know its financial position; every business must have a way of keeping track of who owes money and to whom money is owed; every business must account for parts and supplies or suffer possible loss of sales; a serious business also plans for the future. These activities are sometimes called **management activities** or **institutional activities**, but all businesses must engage in these activities to be true businesses. In data processing these activities are known as **Management Information Systems** and may be roughly but conveniently grouped into seven major categories:

- General Ledger.
- Accounts Receivable.
- Accounts Payable.
- Payroll.
- Order Entry.
- Inventory Control.
- Sales Forecasting/Marketing Research.

Programming for business, in short, is programming in one of these major application areas. To program successfully for business, we must first thoroughly understand these seven business systems.

Let us now briefly discuss what these activities are.

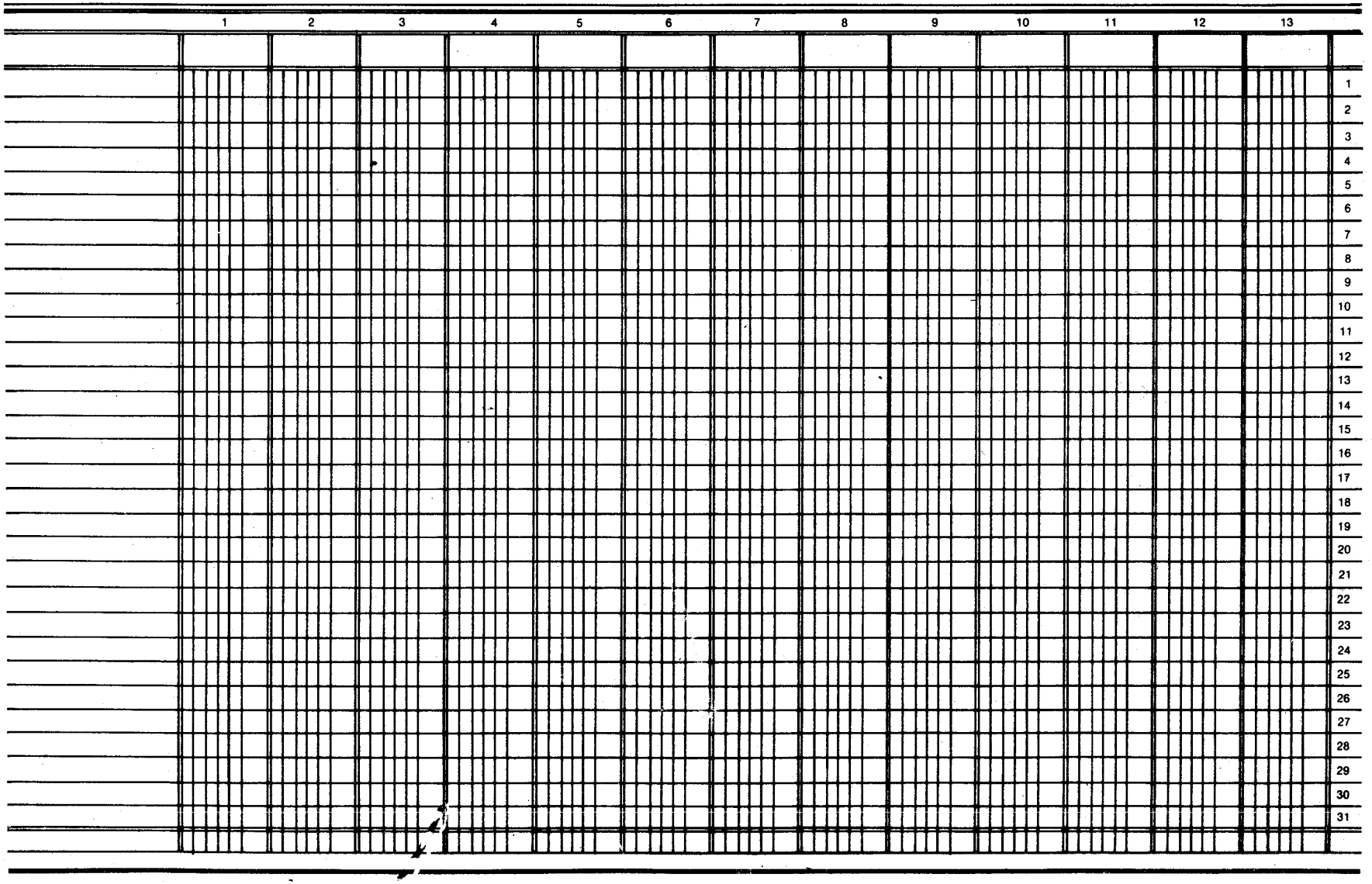
- **General Ledger**

General ledger is the term applied to any bookkeeping activity. A company, for instance, must keep track of all its income and expenses over time. A bookkeeper may lay the income and expenses out in rows and the periods of time in columns. Such income and expenses, usually expressed in dollars and cents, are recorded in table form on accounting sheets that can be purchased from a stationery store (see Figure 1.1); the sheets are kept in a binder called a **ledger**. In a company with automated data processing facilities, these data are maintained on some **secondary storage device** such as magnetic tape or disk. The components that make up the facilities to perform data processing on these numbers constitute the **general-ledger** system.

- Accounts Receivable

Most businesses nowadays transact on credit. Few companies of significant size deal strictly in a cash-and-carry fashion. A retail store may not accept

FIGURE 1.1 A Ledger Sheet



personal checks, but it most likely will accept credit card payments. Credit card payment is a form of credit. So in one way or another a company needs a system to record sales and money due from customers. For companies that extend credit to customers, the time of sale, amount of sale, the credit terms, and the state of customer accounts are all data that must be filed away systematically so that bills and statements can be sent out efficiently and payments can be debited correctly. The data processing facilities handling such activities are called **accounts receivable** systems.

- **Accounts Payable**

Most companies allow customers to defer payment for goods and services rendered by means of credit, recording the debts under accounts receivable. So, too, do companies buy goods and services on credit. An **accounts payable** system records the company's purchases and issues payments at the proper time so the company can maintain its credit standing.

- **Payroll**

All businesses employ people, and employees must be paid. But making payments to employees is not as simple as writing a check. Wages are computed from many personnel data. These include hours or days worked, pay rate, such special events as overtime and holidays or absences due to illness, withholding status, deduction schedules, and bonuses due. And the data may be stored on various media, such as time cards, forms, tape, and disk. A sophisticated system is often needed to perform **payroll** data processing.

- **Order Entry**

Very few businesses sell strictly over-the-counter, where merchandises are immediately delivered. Even a small local flower shop usually takes orders for flowers to be delivered at a later time. Manufacturing companies seldom make direct sales because such concerns typically construct a product to the customer's exact specifications. Keeping track of the orders, establishing priorities and schedules for manufacturing, verifying that finished products are shipped or delivered, all demand a record-keeping system of high accuracy and reliability. Such a system, called **order entry**, is one of the most important in a business: the order-entry system is, in fact, the business' life line.

- **Inventory Control**

A business sells either products or services. In either case a business needs supplies. For a retail business the products are purchased and sold as they are purchased. But for a manufacturing business the products are built from parts, and the products may be diverse in kinds and/or models. Although the parts might be purchased from different suppliers, very often such parts are equivalent and interchangeable. A company usually buys parts and products in large quantities and must keep track of the supply so that a shortage of a particular part will not occur. Sometimes a part cannot be purchased quickly. If the supplied part is custom manufactured, a "lead time" may be necessary to fill an order. Consequently, it is of utmost importance for a business to know the inventory levels accurately at all times; thus, it can react to a possible shortage and take advantage of the best prices in the marketplace. The system for **inventory control** is often one of the most complex systems a company operates.

- **Sales Forecasting/Marketing Research**

Making sure parts do not run out, hiring people to build the products, selling goods and services, and collecting money in payment are not enough to assure a successful business. Competition is keen in the modern business world. A

business must react appropriately and quickly to changes in the conditions of the marketplace. Raw materials may become in short supply. Customers' tastes may change. New products and applications may appear. A company must intelligently "predict" what the market will be like in the near future. A company cannot introduce new products haphazardly without evidence indicating probable success. Even if a product is clearly going to be a hot item, the company must decide whether to place the fate of its new product in the hands of the public immediately or to try and insure success by supplementing with advertising strategies. Questions such as these have no easy answers. In order to make meaningful decisions, a company's managers need relevant information secured through certain analyses. Otherwise, they must operate "by the seat of the pants"; decisions made without the benefit of analysis are no better than pure hunches. Meaningful information must be extracted from raw data by a process known as **sales forecasting and marketing research.**

These seven systems make up what is popularly known as Management Information Systems, or **MIS.** The names of the departments responsible for MIS may differ from one company to another. Different books also may use different names to refer to MIS. Whatever they are called, the essence of the activities is the same. Some companies do not have an accounts receivable department; instead, they have two departments, billing and collections. A hospital may not have a sign posted showing a place for order entry, but patients may inquire about the availability of facilities at a window labeled "Patient Admission." For a school, of course, the "order entry" system is called registration.

THE COMPUTER SYSTEM FOR A SMALL BUSINESS

Certain tools are needed to facilitate the processing of MIS data. In general these tools make up the data processing system, which has three major components:

1. Hardware.
2. Software.
   a. Programs.
   b. Data.
   c. Procedures.
3. People (personnel).

The hardware is the computer. However, there are many types of computers. For an application, the question is always what computer to use. The answer is worth a lot of money to the systems analyst or data processing consultant who earns a living providing the answer. In general, though, some guidelines do exist. Remember, the objective of using a computer for data processing purposes is to achieve accuracy and speed in computation at a reasonable cost. (Factors affecting the accuracy of a computer that result from differences among computer makes and models are not within the scope of this text.) Speed in computation also depends to a large extent on the make of the computer and the proper selection of hardware components. The factors that are truly manageable by the user are the languages and the programs used. We shall discuss the hardware and software separately.

Laypeople most often ask if there is a rule of thumb to the selection of computers. The answer is really no. But in many cases, we do find that large businesses require large computers, medium size businesses use medium-size computers, and small businesses can get by with small computers.

Naturally, we do not have computer categories called large, medium, and small. After all, computers are not pizzas. In proper terminology a large-size computer is known as a **mainframe**, a medium-size computer is called a **mini-**