Walk into any dentist’s office and you will probably find copies of Highlights for Children on various tables and magazine racks. As a child I would nervously bide my time behind the covers of this journal. As you may recall, every issue of Highlights has a cartoon series named “Goofus and Gallant.” The panels of the strip present contrasting views of the lives of two young boys, Goofus, an incorrigible troublemaker, and Gallant, every mother’s favorite child.

Goofus and Gallant are still the same ages they were decades ago, but what if they did grow up? Imagine that they are both in their early thirties and they both happen to be programmers (see Figures 1–1 and 1–2).

Goofus and Gallant truly represent the opposite ends of the selfishness continuum; most people fall somewhere in between these two extremes. Still, these two characters portray two paths that can be followed in many areas of life: the path of reckless selfishness and the path of thoughtful professionalism.

Although you purchased a book on programming style, you are really holding a book on professionalism. Without maintaining a professional demeanor in your programming endeavors, you will not create quality systems. Of course anyone can make something look good, but thoughtfulness and a concern for quality are requirements to make something, such as an application, good on its own. This volume begins the discussion of style with its most important focus: professionalism in programming.

What is professional programming? Is it holding down a job at a software company? Is it having “software engineer” or “programmer/analyst” printed on your business card? While professional programmers often have these things, they alone do not indicate who is truly professional. Professionalism is something that grows and lives inside the heart of the programmer.
Good style flows from this wellspring of quality, affecting not only the programmer’s applications, but his or her entire work ethic.
Programming in Visual Basic is fun because it is so simple to create a full-fledged Windows application. When the Visual Basic development environment first starts up, you are presented with a blank form that represents a fully functional Windows program (just press F5 to prove it). The programmer's job is to mold that core program into a useful and useable application. However, not all Visual Basic programs are created equal. Some applications exhibit the characteristics of a professional program. These characteristics include, but are not limited to, a consistent user interface, clearly defined usage, user documentation, and freedom from bugs. Other applications lack some or all of these features. There are many reasons why poor Visual Basic programs are written.

- **Lack of planning.** Active development of the application begins without careful consideration of the flow of the program, or the needs of the user.
- **Lack of skill or talent.** One or more of the programmers or managers involved in the Visual Basic project are not adequately prepared to meet the challenge of professional Visual Basic application development.
- **Lack of training.** Although those involved in the project are competent enough to complete the project, proper tools, training, and documentation are either unavailable or overlooked.
- **Lack of management.** A disproportionate share of the burden of project management is placed on the shoulders of the programmer(s) instead of on management.
- **Lack of time.** A predetermined implementation schedule fails to take into account the complexities of application development and deployment.
- **Lack of discipline.** A clear set of rules, guidelines, and constraints, whether formal or informal, are placed neither on the management of the Visual Basic project, nor on the activities of the programmers.
- **Lack of scope.** New components of the project are added on a continual basis without proper analysis of needs or time requirements.

Any one of these symptoms can lead to the eventual downfall of a Visual Basic application. When several (or all) of these deficiencies are combined, failure is almost certain. However, these problems can all be prevented, and the chances of success substantially improve by applying the effective methods described in this book.

**Goals of Professional Programming**

Very few developers sit down before the keyboard simply to string random Visual Basic statements together. Rather, there is a purpose, a goal in writing a block of code. The primary goals of professional software development have to do with meeting the needs of various individuals and groups.
Meeting the Needs of the User

All applications are developed for one or more users. Sometimes you write a quick program simply for yourself, to fulfill an immediate processing requirement. At other times you are involved in large-scale, year-long development projects that will eventually be deployed to thousands of users. In both cases, your primary goal is to meet the needs of the user of the application. If you fail to provide the software that the user needs, you fail to deliver quality software, no matter how good it looks.

Computer users have a job to do, and your application automates that job process. The ultimate design goal of an application is to have the user forget that he or she is using a computer program, and instead, simply perform the desired problem-solving task. Your program should be a natural extension of the workflow that is already part of the user's business.

Meeting the Needs of Your Employer

Unless you are working for a nonprofit agency, there is a pretty good chance your company exists in order to make a profit. This holds true for both the self-employed and the multi-billion-dollar corporation. Companies remain profitable by meeting the needs of their customers. You help fulfill both aspects of your company's mission by developing professional software applications.

Meeting your company's needs extends to your managers and coworkers as well. When you develop software in a professional manner, you help to advance the interests of all those around you.

Meeting the Programmer's Needs

Looking after your own personal and professional needs will have a beneficial impact on your software development projects. If you are unhappy with your company or colleagues, or dislike programming because you have to reinvent a methodology every time you start a new project, the quality of your work will likely suffer. However, if you receive satisfaction from your work and enjoy the process of programming high-quality Visual Basic applications, that will shine through brightly in your finished products.

Software development allows for a wonderful mix of challenge and predictability. The challenge comes from stretching your skills into new, untried areas. The predictability comes from the standards that enable you to write consistent, high-quality applications. Some of these standards exist in the syntax of Visual Basic itself. Other standards are created from your own experience, or learned from external sources such as this book. When you expand your programming range by tackling technical challenges, and you learn and apply a rigorous programming methodology, the result will be top-notch programs and applications.
A System for Professional Programming

Who would have thought that typing a few thousand lines of code could satisfy the needs of so many people? Yet that happens every time the professional Visual Basic developer delivers a quality product. So how does one ensure this success on a regular basis? How does a programmer work in a professional manner? Quality programming is accomplished by applying a methodology, a standardized way of attacking complex programming tasks. The method described in The Visual Basic Style Guide has three components, representing the three main divisions of the book.

Structure

When building a home, once you have the foundation in place, you can move on to the structure of the edifice. The same is true of application development. The structure of a software application includes elements such as the logic of the program, the user interface, and supporting documentation. Also, your own personal style plays a large part in how the structure will form over the lifetime of a development project.

The various aspects of application development structure are covered in the first part of this book, Structure. This section begins with this chapter, and continues on through Chapter 5. In these chapters you will encounter numerous guidelines used in the development of all professional software applications.

Chapter 1, Professional Programming as Style. The first chapter, which you are now reading, focuses on the goals and activities of professional software application development.

Chapter 2, Using Declaration. This chapter is dedicated to the importance of object nomenclature and consistency within the source code. The chapter stresses standards and stability throughout all aspects of an application’s source code.

Chapter 3, Commenting and Style. Along with the chapter on documentation (Chapter 5), this chapter discusses the importance of documentation within the source code itself. The professional Visual Basic programmer must demand that adequate documentation appear both within the application source code and in other external forms. The second part of this chapter discusses style, the personality that the developer brings to the source code.

Chapter 4, User Interface Consistency. The user of any software will always notice problems and inconsistencies within the user interface of an application. This chapter stresses the importance of stable user interfaces, and provides examples of when and how to use various interface components.
Chapter 5, Documentation. Besides the software itself, the documentation is the most visible component of any application. Therefore, it needs to be every bit as professional as the source code. This chapter focuses on the need for complete, clear, and high-quality documentation, both for the user community and for the technical staff who develop and maintain the application.

**Foundation**

The foundation of professional software development appears in the heart of the programmer. The core attitudes that help shape the individual also help to shape the programs written by that individual. If you have a professional attitude, the quality of your attitude will show itself in your applications.

While there are many parts to this foundational attitude, the three elements that affect software development the most are discipline, planning, and ethics. These essential attitudes are discussed in detail in the second part of this book, Foundation, encompassing Chapters 6, 7, and 8. These three chapters appear in the middle of this volume to show that they are at the center, at the focus, of the professional way of software development. Much as the planets revolve around the Sun in our solar system, the other activities of professional programming revolve around these three core elements. Without these elements, quality programming is difficult, if not impossible, to attain.

Chapter 6, Discipline. Discipline delves into the mind of the Visual Basic programmer, where all quality development begins. After covering the many reasons why discipline is important to the developer, this chapter discusses various methods of instilling discipline.

Chapter 7, Planning. This chapter discusses the importance of planning in the daily work of the Visual Basic developer. Much time is spent focusing on the needs of the user, and the reasons for adequate planning.

Chapter 8, Ethics. Without ethics, discipline and planning are at the mercy of every whim of the developer, good or bad. This chapter discusses those facets of ethics that pertain to software development, including honesty, quality, humility, and others.

**Standards**

In the building of any modern structure, there are various standards that must be followed to make sure not only that the structure will last a long time, but that it will maintain its high level of quality and value. These standards are used as consistent guides during the construction process. Software development uses comparable standards. This is apparent to the programmer because the syntax of a programming language requires a certain amount of standardization. These intrinsic standards may be enough to write a working program, but they are insufficient to produce the quality or longevity needed to solve
real-world business problems. A full set of application development standards brings consistency and quality to the programming process.

The third and final section of this book, Standards, codifies such a set of standards. Guidelines covering all aspects of the Visual Basic language and its components are examined and stated in Chapters 9 through 12. This section is meant to be handled like a well-worn reference book. Read it through once, then reference its parts frequently during application design and development.

Chapter 9, Declaration Standards. This chapter provides a complete reference (summarized in Chapter 2) of naming conventions to be used within a Visual Basic application.

Chapter 10, Keyword Reference. The largest of the reference chapters, this chapter provides usage and standards information on all of the keywords used in Visual Basic, Visual Basic for Applications, and Visual Basic, Scripting Edition. Each keyword includes recommendations and frequent examples of correct and incorrect usage.

Chapter 11, Control and User Interface Standards. This chapter provides guidelines and sample source code for the most popular user interface controls provided with Visual Basic.

Chapter 12, Database Standards. Most Visual Basic applications exist to present information to and gather information from a user, all of which is stored in a database. This final chapter of the book lists standards that should be used when communicating with databases.

Summary

Fulfilling the needs of everyone involved in a software development project can sometimes seem unattainable. Yet with the proper methods and standards, you will be adequately prepared to meet, and even surpass, those goals. As you move through the remainder of this book, recall often the three primary aspects of professional software development: foundation, structure, and standards.