

## Preface

As this edition was being written, IBM renamed the i5/OS operating system (also known as OS/400) and reconfigured its midrange hardware marketing to concentrate on the consolidated IBM Power Systems line, effectively replacing the traditional AS/400, iSeries, and System i hardware. While the traditional hardware may be gone, the operating system continues on as the IBM i operating system, running on Power Systems servers.

This edition was written using IBM i Release 6.1 as its base release. The CL code examples will work at Release 6.1 or later. Many of the examples will also work at previous i5/OS releases (especially release V5R3 or V5R4), but some may require “workaround” modifications at those releases.

This book can be logically divided into five sections:

1. Introduction to Control Language (Chapters 1–4)
2. Basic CL Programming (Chapters 5–12)
3. Input/Output in a CL Program (Chapters 13–15)
4. Advanced CL Programming (Chapters 16–22)
5. Creating Your Own Commands (Chapters 23–25)

In most cases, an instructor will be able to use the text sequentially to build a complete CL instructional track.

The first section deals generally with CL as the primary interface to IBM i functions. If students have previous experience with the IBM i operating system (or i5/OS or OS/400), this section will serve as review and reinforcement; an instructor most likely will be able to cover this section of the book in a very short time.

The next section, Basic CL Programming (Chapters 5–12), begins to introduce the concept of a CL program and its components. This section also covers many of the capabilities of CL programs, along with discussions about program logic, expressions, program calls, and basic error trapping. After finishing this section, students should be able to read a CL program, identify its function, and write a simple CL program.

During the study of Chapters 5–12 and throughout the rest of the book, students often will find it useful to refer to the appendices. Appendix A is a “quick reference” to many CL commands used most often in CL programs. For students who have no experience with IBM i programming tools, Appendix B introduces Programming Development Manager (PDM), Source Entry Utility (SEU), and WebSphere Development Studio Client (WDS*c*).

To further build the students’ knowledge of CL, the third section (Chapters 13–15) covers CL’s various input and output capabilities — for example, parameters, files, data areas, and system values. After finishing this section of the text, a student should be able to write a CL program to perform a relatively complex function.

Although the next section (Chapters 16–22) is presented as “advanced,” several topics in this section should be part of even a minimal CL course of study. In particular, Chapters 16, 17, and 19, which cover messages, error handling, file overrides, and APIs, should always be included. Chapter 22 discusses fundamental changes that the Integrated Language Environment (ILE) introduces to

the IBM i execution model; this information is relevant not only to CL and should be covered thoroughly so that the student understands how ILE changes some of the traditional IBM i concepts.

Chapters 23–25 discuss the construction of user-defined commands; a knowledge of this facility will give students an advantage in the IBM i job market, and its coverage is well worth the time spent studying it. If the length of a course does not permit inclusion of user-defined commands, the instructor should encourage self-study in this area.

The end of each chapter includes a list of key terms defined in the chapter and in the glossary. These terms are for class discussion to ensure that students grasp the concepts presented in the chapter. Some of the review questions also lend themselves more to classroom discussion than to individual testing.

Some of the programming assignments at the end of the chapters build on previous assignments — an important consideration if all the programming assignments in a chapter cannot be completed in the computer lab. With this constraint in mind, instructors should choose those assignments that they feel will be most beneficial to the students.