

Bibliographie

P. Massie: Programming IBM Assembly Language & Instructor's Manual To Accompany "Programming IBM Assembly Language", XIII+342 & I+149 pages, Burgess Communications 1985.

The first book is a comprehensive introduction to IBM assembler language programming. Unlike many books on the subject, it is not a reference manual. "It is designed to carefully balance concepts with practical applications at a level that students will find most useful. Normally students will have completed at least one course in programming before the assembler language course, and will be familiar with many of the basic concepts involved. With this book, experience is not absolutely necessary, since programming steps are presented fully in a logical, building-block sequence. To help build confidence and mastery of the central concepts, students are encouraged to run programs early".

The first book is divided into six parts and twenty-four chapters.

PART ONE, "INTRODUCTION", covers the basic concepts necessary for assembler language programming, but different from high-level languages. Operating system interfaces can be a complex topic, and the initial presentation in this section is done in a simple fashion. This allows the student to begin writing programs quickly, without a full understanding of these interfaces. Dumps are initially used to obtain output from the programs, so the student becomes familiar with reading and understanding dumps at an early stage.

PART TWO, "REGISTER-BASED INSTRUCTIONS", presents register instructions in a simple format designed for easy comprehension by the beginner student. In addition, this section includes a chapter on the simpler methods of developing loops in assembler language. Although structured programming involves much more than simply reducing branch statements in programs, it is important to introduce the basic concepts of structured programming early.

PART THREE, "CHARACTER PROCESSING", covers the different data formats available, the normal I/O macros, and simple character-processing instructions. With the building-blocks of the earlier parts of the book, students should be ready to handle these more complex subjects. The I/O chapter does not present I/O macros exhaustively, but at a level such that students can run programs.

PART FOUR, "PACKED DECIMAL", covers the packed decimal instructions. This is a major family of arithmetic-processing instructions different in concept from the fixed-point binary instructions covered earlier.

PART FIVE, "GENERAL TOPICS", is a carefully-selected collection of chapters on distinct subjects, which can be covered in connection with other chapters, or completely omitted. Many of these topics are advanced, and can be assigned in accordance with the level and interest of the student.

PART SIX, "ADVANCED TOPICS", covers macros and conditional assembly. The intention is to provide an overview of these advanced features of the IBM assembler.

The APPENDICES also contain a sample JCL, the complete instruction set, and a glossary of selected terms necessary for assembler language programming.

To help students master concepts and to reinforce the logical building-block structure of the book, each chapter contains a series of important learning aids.

- **KEY CONCEPTS** opens each chapter and focus student attention on what is to be learned.

- **INTRODUCTIONS** provide an overview of each chapter and summarize their central concepts.
- **KEY TERMS** are presented in boldface when introduced, and full definitions are provided in the glossary.
- **EXAMPLES** are presented in colored boxes to ensure student engagement.
- **EXERCISES** are introduced throughout each chapter to help students assess their mastery of the material.
- **PROGRAMS**, both partial and complete sample listings, are set off by horizontal rules to focus student attention.
- **CHAPTER SUMMARIES** review terms and concepts, and provide students with study devices in preparation for classroom discussions and examinations.
- the **GLOSSARY** of terms of the end of the book fully defines terms, and provides an additional study aid for mastery of the material.

The second book, the Instructor's Manual, offers complete support for the instructor's classroom presentations. The manual provides the instructor with the following aids:

- Complete chapter summaries
- Chapter objectives
- Answers to chapter exercises
- Multiple choice and true/false examination questions.

The Instructor's Manual provides as much assistance as possible to the instructor of the course. These very well-written books can be recommended to students, instructors and everyone interested in programming IBM assembly language.

K. Dévényi (Szeged)

D. F. Stubbs and N. W. Webre, Data Structures with Abstract Data Types and Pascal, XVIII + 459 pages, Brooks/Cole Publishing Company, Monterey, California, 1985.

"This text represents a fresh approach to a first course in data structures. During the past several years, researchers have developed a new method of designing data structures, the new forms of which are called abstract data types. In the original research papers, abstract data types are quite formal and unsuited to the beginning student. When the formality is removed, however, there remains a basic and easily understood essence.

Our major challenge in writing this book was to make the notion abstract data types an integral part of the study of data structures. At the end of the course, we want students to have knowledge of 'classical' data structures, as well as skills in the use of abstraction, specification, and program construction using modules, or packages. These skills prepare students to make use of the data abstraction facilities of languages such as Ada, Modula 2, and Mesa. They are also an excellent complement to the techniques of top-down design and structured programming. This text prepares students for advanced courses in which the methods are more formal. In addition, it will serve as an important first step toward object-oriented programming."

I certainly recommend the book as a possible text for a first course in data structures.

Gy. Horváth (Szeged)

S. Grier, Pascal for the 80s, XVI + 540 pages, Brooks/Cole Publishing Company, Monterey, California, 1985.

This book is designed to serve as an introductory text in programming for college students. It does not attempt to teach a complete knowledge of Pascal, nor to make any student a mature or expert programmer. Advanced topics in Pascal as sets, pointers and recursion are briefly discussed. This text provides an understanding of very simple algorithms. The main emphasis is on the concept of teaching problem-solving in parallel with the programming language.

The book is recommended for people who are interested in teaching programming at a non-academic level.

Gy. Horváth (Szeged)

Natural language parsing. Psychological, computational, and theoretical perspectives (Edited by D. R. Dowty, L. Karttunen, A. M. Zwicky) XIII+413 pages, Cambridge University Press, 1985.

The volume is a collection of papers written by leading researchers in experimental psychology, theoretical linguistics, and artificial intelligence. The contributions are: Introduction (Lauri Karttunen and Arnold M. Zwicky), Measuring syntactic complexity relative to discourse context (Alice Davison and Richard Lutz), Interpreting questions (Elisabet Engdahl), How can grammars help parsers? (Stephen Crain and Janet Dean Fodor), Syntactic complexity (Lyn Frazier), Processing of sentences with intrasentential code switching (Aravind K. Joshi), Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? (Aravind K. Joshi), Parsing in functional unification grammar (Martin Kay), Parsing in a free word order language (Lauri Karttunen and Martin Kay), A new characterization of attachment preferences (Fernando C. N. Pereira), On not being led up the garden path: the use of context by the psychological syntax processor (Stephen Crain and Mark Steedman), and Do listeners compute linguistic representations? (Michael K. Tanenhaus, Greg N. Carlson, and Mark S. Seidenberg).

The book is warmly recommended to linguists, psycholinguists and computer scientists.

F. Gécseg (Szeged)

Victor J. Law: Standard Pascal; an Introduction to Structured Software Design, XVII+558 pages, Wm. C. Brown Publishers, Dubuque Iowa, 1985.

"It is well recognized that an introductory course in computing should do more than teach the syntax of a high-level language. The course described as CS1, Introduction to Programming Methodology, in CACM, October 1984, lists the important topics that should be covered in early computing courses. This text melds these non-language issues and the Pascal language into a cohesive presentation of modern programming methodology for beginning students."

This text is designed to satisfy two major objectives. One goal is to introduce modern principles of programming, which can be used for small and large programming projects. The other objective is to introduce the Pascal language as a vehicle for implementing algorithms designed with the software life cycle approach.

Chapters are segmented into three parts. The first part introduces new concepts. A sample problem is then posed, a specification is written, and an algorithm design is presented. A relatively new algorithm design tool, the structured chart, is used for all of the designs. The second part covers any new Pascal features required to implement the new concepts as introduced in the first part. The third part contains complete programming examples where the new topics are featured. Each example includes requirements, specification, algorithm, design, Pascal code, and testing. Each chapter ends with review questions, self-test exercises, and programming problems. Advanced problems are excluded. The appendixes include, among others, answers to selected self-tests, a summary of Pascal syntax using syntax diagrams, and a comparison of common Pascal dialects.

This book is recommended as a text for an introductory course in computing.

Gy. Horváth (Szeged)

N. Wirth: Programmieren in Modula-2, XIV+220 Seiten, Springer-Verlag, Berlin Heidelberg New York Tokyo, 1985.

Das Buch ist die Übersetzung der englischen Original-Ausgabe: Programming in Modula 2, Third corrected Edition, Springer-Verlag, 1985.

Als Handbuch für die Programmierung in Modula-2 überdeckt der Text praktisch alle Eigenschaften dieser Sprache. Teil 1 umfaßt die elementaren Begriffe Variable, Ausdruck, Zuweisung, bedingte und Wiederholungs-Anweisung sowie die Datenstruktur des Arrays. Teil 2 führt in das wichtige Konzept der Prozeduren bzw. Unterprogramme ein. Beide, Teil 1 und Teil 2 umfassen im Wesentlichen den Stoff eines Einführungskurses in die Programmierung. Teil 3 befaßt sich mit Datentypen und Strukturen. Dies entspricht im Kern dem Inhalt eines weiterführenden Programmierkurses. Teil 4 führt den Begriff des Moduls ein, eines fundamentalen Konzeptes sowohl für den Entwurf großer Programmsysteme als auch für das Arbeiten in einem Team. Als Beispiel für Module werden einige häufig verwendete Hilfsprogramme für Ein- und Ausgabe dargestellt. Teil 5 schließlich beschreibt Möglichkeiten der Systemprogrammierung, Gerätebehandlung und der Multiprogrammierung. Weiterhin werden praktische Hinweise gegeben, wie und wann die einzelnen Hilfsmittel einzusetzen

sind. Dies ist als Richtschnur für den Erwerb eines anständigen Stils der Programmierung und der Systemstrukturierung zu verstehen.

Empfohlen werden kann das Buch all denen, die an Programmierung in irgendeiner Weise interessiert sind.

J. Csirik (Szeged)

H. Bunke: Modellgesteuerte Bildanalyse, VIII + 301 Seiten (Reihe: Leitfäden der angewandten Informatik), B. G. Teubner, Stuttgart 1985.

Das Buch befasst sich mit der automatischen Analyse von Bildern und Bildfolgen mittels eines Digitalrechners. Es gliedert sich in zwei Teile: im ersten Teil werden die Grundlagen der wissenschaftlichen Bildanalyse angegeben, und der zweite Teil beschäftigt sich mit einem speziellen wissenschaftlichen Bildanalyse-System, das an der Universität Erlangen-Nürnberg entwickelt wurde.

Bei der wissenschaftlichen Bildanalyse (im ersten Teil des Buches) werden drei größere Gebiete angesprochen:

- Methoden zur Extraktion elementarer Bestandteile,
- Wissensdarstellung,
- Wissensnutzung.

Das Material jedes einzelnen Gebietes wird durch einführende Beispiele und durch einen sehr breiten und ausführlichen Literaturüberblick ergänzt. Dies und der logische Aufbau des Textes machen diesen Teil des Buches zu einer beispielhaften Einführung in die wissenschaftliche Bildanalyse.

Im zweiten Teil der Arbeit wird ein wissenschaftliches System zur Analyse nuklearmedizinisch gewonnener Bildfolgen des menschlichen Herzens detailliert vorgestellt. Das Hauptziel dabei ist: die automatische Ableitung einer Diagnose aus einer Bildfolge. Das System ist modular aus den folgenden Komponenten aufgebaut: Modell, Instanzen, Methoden, Kontrolle und Dialog. Dabei werden die grundlegenden Ideen ausführlich beschreiben und das Konzept der Arbeit übersichtlich vorgestellt. Auch die ersten Ergebnisse des klinischen Einsatzes sind gegeben.

Das Buch passt sehr gut zu den Zielen der Reihe "Leitfäden der angewandten Informatik". Ich empfehle es sowohl für Fachleute eines anderen Gebietes als auch für Informatiker, die auf diesem Gebiet tätig werden wollen.

J. Csirik (Szeged)

Varrel C. Grout: Programming with BASIC, XIII + 362 pages, Wm. C. Brown Publishers, Dubuque, Iowa, 1985.

This book is written for high-school graduates who have had no previous experience with computers or BASIC programming, but it is useful even for advanced programmers and teachers.

The book is more than simply a BASIC text-book. It provides information about how to define problems and design algorithms in order to solve them efficiently.

The book consists of 13 chapters. The first 3 chapters provide a textual and pictorial introduction to computers and programming. Fundamental BASIC statements and structures are described in the following five sections. Chapters 9 through 13 can be used to provide information about data file processing, printed output designing, arrays and special matrix statements. Each section contains some exercises and useful self-assessment tests. The author emphasizes the advantages of structured programming and the efficiency of modular design. Indeed, every program example is designed and developed in a structured way.

The author's aim was to give information about the fundamentals of modern problem-solving, such as the stepwise refinement procedure for designing algorithms, modularity, etc.

An other advantage of the book is the choice of the form of BASIC language. The Microsoft BASIC used in the book is a widespread form of BASIC language.

L. Czinkóczy (Szeged)