**PASCAL**

**PROGRAMMING**

Dr. DOGAN IBRAHIM

BSc MSc PhD MIEEE MBCS MIEE CEng

NEAR EAST UNIVERSITY

Nicosia, Cyprus 1993

FIRST EDITION



•.. --

(C)1993

First published 1993

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the Author.

Printed & Bound in the

Turkish Republic Of Northern Cyprus, Nicosia.

**PREFACE**

This bookis aimed for the second year University students studying Computer Programming, Computer Science, Computer Information Systems, or Computer Engineering. The topics in the book have successfully been thought to the second year Computer Engineering students at the Near East University. It is recommended that the book should be completed in one semester.

The material in the book has been prepared with the SVS PASCAL compiler in mind, although the book could easily be used in teaching any PASCAL programming language. Some previous knowledge of another programming language would be useful, although not essential.

All the programs in the book have been tested and run on the UNISYS 6000/35 series computer of the Near East University, running the UNIX operating system.

Exercises are provided at the end of every chapter to help the students to practice their knowledge. The recommended method of study is such that the students should have access to an online computer system, or preferably to a personal computer with the PASCAL compiler.

I am grateful to my wife who checked the entire manuscript for errors and also put the manuscript into a form which can easily be understood.

Dr. Dogan Ibrahim January 1993 - Nicosia.

**Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PREFACE** | |  |  | **3** |
| **1.** | **INTRODUCTION** | | | **9** |
|  |  |  | . |  |
|  |  |  | . |  |
| **2.** | **DATA TYPES** | |  | **11** |
|  | 2.1 | The Alphabet | |  |
|  | 2.2 | Variable Names | |  |
|  | 2.3 | Constants | |  |
|  | 2.4 | Variables | | **14** |
|  |  | 2.4.1 | Standard Variable Types | 15 |
|  |  | 2.4.2 | User Defined Data Types | 17 |
|  | 2.5 | Assignment Statements | | 18 |
|  | 2.6 | Arithmetic Expressions | | 19 |
|  |  | 2.6.1 | Hierarchy Of Operations | 20 |
|  |  | 2.6.2 | Numbers | 23 |
|  | 2.7 | Exercises | | 25 |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3.** | **STRUCTURE OF A PASCAL PROGRAM** | | | **29** |
|  | 3.1 | Comments In PASCAL programs | | 30 |
|  | 3.2 | Displaying Data | | 32 |
|  | 3.3 | Output Field Width | | 36 |
|  |  | 3.3.1 | Displaying Integer Numbers | 36 |
|  |  | 3.3.2 | Displaying Real Numbers | 39 |
|  | 3.4 | Input In PASCAL | | 45 |
|  |  | 3.4.1 | Character Data | 50 |
|  | 3.5 | Exercises | | 53 |

**CONTROL STRUCTURES**

**58**

**4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 4.1 | WHILE Statement |  | 62 |
|  | 4.2 | REPEAT Statement |  | 69 |
|  | 4.3 | IF Statement |  | 73 |
|  | 4.4 | FOR Statement |  | 76 |
|  | 4.5 | CASE Statement |  | 81 |
|  | 4.6 | GOTO Statement |  | 87 |
|  | 4.7 | Reading A String Of Characters | | 89 |
|  | 4.8 | Exercises | . | 91 |
| 5. | FUNCTIONS | |  | 96 |
|  | 5.1 | Built-in Functions |  | 96 |
|  | 5.2 | User Defined Functions |  | 106 |
|  | 5.3 | Exercises |  | 113 |
| 6~ |  |  |  | 115 |
|  | 6.1 | Local And Global Variables |  | 120 |
|  | 6.2 | Exercises |  | 123 |
| 7. | ARRAYS | |  | 125 |
|  | 7.1 | Multi-Dimensional Arrays |  | 131 |
|  | 7.2 | Exercises |  | 139 |
| 8. | ENUMARATED TYPES | |  | 141 |
|  | 8.1 | Passing Arrays To Procedures | | 144 |
|  | 8.2 | Exercises |  | 146 |
| 9. | RECORDS | |  | 148 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10. | STRINGS | | 151 |  |
|  |  |  |  | ~ -r |
|  | 10.1 | Strings As Arrays Of Characters | 151 | } |
|  | 10.2 | Strings As Packed Arrays | 154 |  |
|  | 10.3 | Using Record Structures | 157 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10.4 | Using the String Data Type | | | |  |  |  | .. | | 161 |
|  | 10.5 | String Functions | |  |  |  |  |  | ' | | 163 |
|  |  | 10.5.1 | LENGTH |  |  | -,,\_ &--.... | | | | I | 163 |
|  |  | 10.5.2 CONCAT | |  |  |  |  |  |  |  | 164 |
|  |  | 10.5.3 DELETE | |  |  |  |  | ~ |  |  | 165 |
|  |  | 10.5.4 | INSERT |  |  |  |  | ~ |  |  | 166 |
|  |  |  |  |  |  |  | \.c - |  |
|  |  | 10.5.5 COPY | |  |  |  |  |  |  |  | 168 |
|  |  | 10.5.6 | POS |  |  |  |  |  | .•. | | 169 |
|  | 10.6 | Exercises | |  |  |  |  |  |  |  | 171 |
| 11. | FILES |  |  |  |  |  |  |  |  |  | 173 |
|  | 11. 1 | Creating A Text File | | |  |  |  |  |  |  | 174 |
|  | 11.2 | Closing A File | |  |  |  |  |  |  |  | 175 |
|  | 11.3 | Opening An Existing File | | | |  |  |  |  |  | 176 |
|  | 11.4 | Reading Data From A File | | | |  |  |  |  |  | 178 |
|  | 11.5 | Exercises | |  |  |  |  |  | vt:. - I | | 189 |
| 12. | FUNCTIONS REFERENCE | | | |  |  |  |  |  |  | 191 |
|  | Abs |  |  |  |  |  |  |  |  |  | 192 |
|  | Arctan |  |  |  |  |  |  |  |  |  | 193 |
|  |  |  |  |  |  |  |  |  | " . ' | |
|  | Chr |  |  |  |  |  |  |  | . , | | 194 |
|  | Con cat |  |  |  |  |  |  |  |  |  | 195 |
|  | Copy |  |  | l • ""'· : \. | |  | • |  | - l• | | 196 |
|  |  |  |  |  |  | - \_, | |  |
|  | Cos |  | ! |  |  |  |  |  |  |  | 197 |
|  |  |  |  |  |  |  | ' |  |  |  |  |
|  | Delete |  |  |  |  |  |  |  |  |  | 198 |
|  | Eoln |  |  |  |  |  |  |  | *r* |  | 199 |
|  | Exit |  |  |  |  |  |  |  |  |  | 200 |
|  | Exp |  |  |  |  |  |  |  | ,~J;. | | 20 |
|  | Halt |  |  |  | ... |  |  |  |  |  | 202 |
|  |  |  |  |  | ~ . ~ | |  |  |  |  |  |
|  |  |  |  |  |  | ' |  |  |  |  |  |
|  |  |  |  | 6 |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Insert | 203 |
| Length | 204 |
| Ln | 205 |
| Odd | 206 |
| Pos | 207 |
| Pred | 208 |
| Round | 209 |
| Pwroften | 210 |
| Sin | 211 |
| Sqr | 212 |
| Sqrt | 213 |
| Su cc | 214 |
| Trunc | 215 |

|  |  |  |
| --- | --- | --- |
| 13. | APPENDIX A | 216 |
|  | PASCAL RESERVED WORDS |  |
| 14. | APPENDIX B | 218 |
|  | PASCAL STORAGE ALLOCATION |  |
| 15. | APPENDIXC | 220 |
|  | USING THE SVS PASCAL COMPILER |  |
| 16. | APPENDIX D | 225 |
|  | ASCII CHARACTER CODES |  |

17.

APPENDIX E BIBLIOGRAPHY

227

18.

APPENDIX F

GLOSSARY OF COMPUTING TERMS

22~

.....

*c* -. t.



8

**1**

**INTRODUCTION**

PASCAL is one of the popular programming languages designed in the late 1960s. This programming language was first designed by Professor Niklaus Wirth of the Eidenissiche Technische Hocheschule of Zurich, Switzerland. PASCAL is not an acronym for anything and it is named after Blaise Pascal, the 17th century philosopher and mathematician.

For historical reasons, FORTRAN, COBOL and BASIC are more widely used high-level languages than PASCAL. FORTRAN, designed in the early 1950s, was the first widely used high-level computer programming language and this language is very useful for engineering applications. COBOL, developed in the late 1950s, was intended for business and data processing applications. BASIC is an easy to learn programming language which is available on nearly all computers, including the smallest types of home computers. Although the concepts of most prograrnrninq languages are the same, PASCAL offers some useful concepts not found in other programming languages.

Although the early PASCAL was only available on the mainframe computers, it is now very commonly used on mini computers and personal computers (PCs}.

PASCAL is one of the derivatives of the well known structured programming language known as ALGOL-60. PASCAL provides the user the ability to create his or her data types. This is a very powerful utility enabling the prngrammers to create very complex data types and data structures from simple data types.

PASCAL is a compiled language. PASCAL source code (or the PASCAL program} is written using a program editor. The source code is then compiled using a PASCAL compiler. The compiler generates an executable machine code which is then run on the host computer.

9

**1**

**INTRODUCTION**

PASCAL is one of the popular programming languages designed in the late 1960s. This programming language was first designed by Professor Niklaus Wirth of the Eidenissiche Technische Hocheschule of Zurich, Switzerland. PASCAL is not an acronym for anything and it is named after Blaise Pascal, the 17th century philosopher and mathematician.

For historical reasons, FORTRAN, COBOL and BASIC are more widely used high-level languages than PASCAL. FORTRAN, designed in the early 1950s, was the first widely used high-level computer programming language and this language is very useful for engineering applications. COBOL, developed in the late 1950s, was intended for business and data processing applications. BASIC is an easy to learn programming language which is available on nearly all computers, including the smallest types of home computers. Although the concepts of most prograrnrninq languages are the same, PASCAL offers some useful concepts not found in other programming languages.

Although the early PASCAL was only available on the mainframe computers, it is now very commonly used on mini computers and personal computers (PCs}.

PASCAL is one of the derivatives of the well known structured programming language known as ALGOL-60. PASCAL provides the user the ability to create his or her data types. This is a very powerful utility enabling the prngrammers to create very complex data types and data structures from simple data types.

PASCAL is a compiled language. PASCAL source code (or the PASCAL program} is written using a program editor. The source code is then compiled using a PASCAL compiler. The compiler generates an executable machine code which is then run on the host computer.

9

The PASCAL compiler described in this book is the Silicon Valley Software (SVS) compiler. This compiler is based upon the ANSI standard and it has a variety of nonstandard extensions to enable users to carry out more complex programming tasks. SVS PASCAL compiler is available on the Motorola 68020 and the Intel 386 based systems.

The format of a PASCAL program is as follows:

**PROGRAM** name (input.output); declarations; declarations;

statements; statements;

**BEGIN**

**END.**

A PASCAL program begins with the keyword **PROGRAM,** followed by a program name. Every line in a PASCAL program is terminated with a semicolon. There could be a number of declarations after the program name. The actual executable code starts after the keyword **BEGIN.** The lines after BEGIN are valid executable PASCAL programming lines. Every PASCAL program is terminated with the keyword **END,** followed by a dot.

As an example, the following program (named SIMPLE), displays the message "A SIMPLE PROGRAM" on the screen:

**PROGRAM** SIMPLE(output); **BEGIN**

writeln("A SIMPLE PROGRAM");

As we shall see in the next chapters. stat~-\*~ on the screen and it is equivalent to the **P**

**END.**