

THOMSON  
COURSE TECHNOLOGY



# Simple Program Design A Step-by-Step Approach

Fifth Edition

Lesley Anne Robertson

# Contents

<b>Preface</b>	xi
----------------	----

## 1 Program design

*Describes the steps in the program development process, and introduces current program design methodologies, procedural and object-oriented programming, algorithms, pseudocode and program data.*

1.1	Steps in program development	2
1.2	Program design methodology	4
1.3	Procedural versus object-oriented programming	5
1.4	An introduction to algorithms and pseudocode	6
1.5	Program data	7
	Chapter summary	10

## 2 Pseudocode

*Introduces common words, keywords and meaningful names when writing pseudocode. The Structure Theorem is introduced, and the three basic control structures are established. Pseudocode is used to represent each control structure.*

2.1	How to write pseudocode	12
2.2	Meaningful names	15
2.3	The Structure Theorem	15
	Chapter summary	17

## 3 Developing an algorithm

*Introduces methods of analysing a problem and developing a solution. Simple algorithms that use the sequence control structure are developed, and methods of manually checking the algorithm are determined.*

3.1	Defining the problem	19
3.2	Designing a solution algorithm	23
3.3	Checking the solution algorithm	25
	Chapter summary	33
	Programming problems	33

## 4

**Selection control structures**

*Expands the selection control structure by introducing multiple selection, nested selection and the case construct in pseudocode. Several algorithms, using variations of the selection control structure, are developed.*

4.1	The selection control structure	36
4.2	Algorithms using selection	40
4.3	The case structure	48
	Chapter summary	51
	Programming problems	51

## 5

**Repetition control structures**

*Develops algorithms that use the repetition control structure in the form of DOWHILE, REPEAT...UNTIL, and counted repetition loops.*

5.1	Repetition using the DOWHILE structure	55
5.2	Repetition using the REPEAT...UNTIL structure	64
5.3	Counted repetition	67
	Chapter summary	69
	Programming problems	70

## 6

**Pseudocode algorithms using sequence, selection and repetition**

*Develops algorithms to solve eight simple programming problems using combinations of sequence, selection and repetition constructs. Each problem is properly defined; the control structures required are established; a pseudocode algorithm is developed; and the solution is manually checked for logic errors.*

6.1	Eight solution algorithms	73
	Chapter summary	85
	Programming problems	85

## 7

**Array processing**

*Introduces arrays, operations on arrays, and algorithms that manipulate arrays. Algorithms for single and two-dimensional arrays, which initialise the elements of an array, search an array and write out the contents of an array, are presented.*

7.1	Array processing	89
7.2	Initialising the elements of an array	92
7.3	Searching an array	94
7.4	Writing out the contents of an array	96
7.5	Programming examples using arrays	97
7.6	Two-dimensional arrays	101
	Chapter summary	105
	Programming problems	105

## 8 First steps in modularisation

*Introduces modularisation as a means of dividing a problem into subtasks. Hierarchy charts and parameter passing are introduced, and several algorithms that use a modular structure are developed.*

8.1	Modularisation	110
8.2	Hierarchy charts or structure charts	114
8.3	Steps in modularisation	116
8.4	Programming examples using modules	118
	Chapter summary	129
	Programming problems	130

## 9 General algorithms for common business problems

*Develops a general pseudocode algorithm for four common business applications. All problems are defined; a hierarchy chart is established; and a pseudocode algorithm is developed, using a mainline and several subordinate modules. The topics covered include report generation with page break, a single-level control break, a multiple-level control break and a sequential file update program.*

9.1	Program structure	134
9.2	Report generation with page break	135
9.3	Single-level control break	137
9.4	Multiple-level control break	141
9.5	Sequential file update	146
	Chapter summary	152
	Programming problems	153

## 10 Communication between modules, cohesion and coupling

*Introduces communication between modules and develops algorithms that pass parameters between modules. Module cohesion and coupling are introduced, several levels of cohesion and coupling are described, and pseudocode examples of each level are provided.*

10.1	Communication between modules	160
10.2	Programming examples using parameters	165
10.3	Module cohesion	174
10.4	Module coupling	180
	Chapter summary	185
	Programming problems	185

<b>1 1</b>	<b>An introduction to object-oriented design</b>	
	<i>Introduces object-oriented design, classes and objects, attributes, responsibilities, operations, accessors and mutators, and information hiding. The steps required to create an object-oriented solution to a problem are provided and solution algorithms developed.</i>	
	11.1 Introduction to object-oriented design	192
	11.2 Public and private access methods	198
	11.3 Steps in creating an object-oriented solution	203
	11.4 Programming examples using object-oriented design	204
	Chapter summary	212
	Programming problems	212
<b>1 2</b>	<b>Object-oriented design for more than one class</b>	
	<i>Introduces relationships between classes including association, aggregation, composition and generalisation. Introduces a simplified UML language, describes polymorphism and operation overriding and lists the steps required to create an object-oriented design for a problem with more than one class.</i>	
	12.1 Further object-oriented design	216
	12.2 Steps in creating an object-oriented solution using more than one class	221
	12.3 Programming examples using more than one class	222
	Chapter summary	238
	Programming problems	238
<b>1 3</b>	<b>Object-oriented design for multiple classes</b>	
	<i>Expands an object-oriented solution to cater for multiple classes, inheritance and polymorphism and introduces interface and GUI design.</i>	
	13.1 Object-oriented design for multiple classes	242
	13.2 Interface and GUI objects	256
	Chapter summary	259
	Programming problems	259

## **Appendix 1 Flowcharts**

*Introduces flowcharts for those students who prefer a more graphic approach to program design. Algorithms that use a combination of sequence, selection and repetition are developed in some detail.*

Introduction to flowcharts	264
The three basic control structures	265
Simple algorithms that use the sequence control structure	268
Flowcharts and the selection control structure	272

Simple algorithms that use the selection control structure	274
The case structure expressed as a flowchart	280
Flowcharts and the repetition control structure	282
Simple algorithms that use the repetition control structure	283
Further examples using flowcharts	291
Flowcharts and modules	315

## **Appendix 2** Special algorithms

*Contains a number of algorithms that are not included in the body of the textbook and yet may be required at some time in a programmer's career.*

Sorting algorithms	327
Dynamic data structures	330

## **Appendix 3**

Translating psuedocode into computer languages: quick reference chart	335
--	-----

<b>Glossary</b>	342
-----------------	-----

<b>Index</b>	349
--------------	-----