
Peter Sestoft

Henrik I. Hansen

C# Precisely

Version 1.04.0 of 2005-12-27

The MIT Press
Cambridge, Massachusetts
London, England

© 2004 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

This book was set in Times by the authors using L^AT_EX.

Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Sestoft, Peter.

C# precisely / Peter Sestoft and Henrik I. Hansen.

p. cm.

Includes bibliographic references and index.

ISBN 0-262-69317-8 (pbk.: alk. paper)

1. C# (Computer program language) I. Hansen, Henrik I. II. Title.

QA76.73.C154S47 2004

005.13'3—dc22

2004048288

10 9 8 7 6 5 4 3 2

Contents

Preface	ix
Notational Conventions	x
1 Compiling, Loading and Executing C# Programs	2
1.1 Source Files and Compilation	2
1.2 Execution	2
2 Names and Reserved Names	4
3 C# Naming Conventions	4
4 Comments and Program Layout	4
5 Data and Types	6
5.1 Value Types and Simple Types	6
5.2 Reference Types	8
5.3 Conversion	10
6 Variables, Parameters, Fields, and Scope	14
6.1 Scope of Variables, Parameters and Members (Including Fields)	14
6.2 Default Values and Definite Assignment	14
7 Strings	16
7.1 String Methods	18
7.2 String Formatting	20
8 String Builders	24
9 Arrays	26
9.1 One-dimensional Arrays	26
9.2 Multi-dimensional Arrays	28
9.3 Class Array	30
10 Classes	32
10.1 Class Declarations and Class Bodies	32
10.2 Class Modifiers abstract, sealed, static	34
10.3 Member Access Modifiers private, protected, internal, public	34
10.4 Subclass, Base Class, Inheritance and Hiding	36
10.5 Field Declarations in Classes	38
10.6 Constant Declarations in Classes	38
10.7 Method Declarations	40
10.8 Method Modifiers static, new, virtual, override, sealed, abstract	42
10.9 Constructor Declarations	44

10.10 Static Field Initialization and the Static Constructor	44
10.11 Member Classes or Nested Classes	46
10.12 Class Access Modifiers	46
10.13 Property Declarations	48
10.14 Indexer Declarations	48
10.15 Operator Overloading	50
10.16 User-Defined Conversion Operators	52
10.17 Events	52
11 The Machine Model: Stack, Heap, and Garbage Collection	54
11.1 Class and Object versus Struct Type and Struct Value	54
12 Expressions	56
12.1 Table of Expression Forms and Predefined Operators	56
12.2 Arithmetic Operators	58
12.3 The checked and unchecked Operators	58
12.4 Logical Operators	58
12.5 Bitwise Operators and Shift Operators	60
12.6 Comparison Operators	60
12.7 Assignment Expression	62
12.8 Conditional Expression	62
12.9 Object Creation Expression	64
12.10 Struct Value Creation Expression	64
12.11 Instance Test Expression	64
12.12 Instance Test and Cast Expression	64
12.13 Field Access Expression	66
12.14 The Current Object Reference <code>this</code>	66
12.15 Method Call Expression	68
12.16 Property Access Expression	76
12.17 Indexer Access Expression	78
12.18 Type Cast Expression	80
12.19 The <code>typeof</code> operator	80
12.20 Anonymous Method Expression (C# 2.0)	82
13 Statements	84
13.1 Expression Statement	84
13.2 Block Statement	84
13.3 The Empty Statement	84
13.4 Declaration Statement	84
13.5 Choice Statements	86
13.6 Loop Statements	88
13.7 Returns, Labeled Statements, Exits, and Jumps	92
13.8 Throwing and Catching Exceptions	94
13.9 The checked and unchecked Statements	98

13.10 The using Statement	98
13.11 The lock Statement	98
13.12 The yield Statement and Iterators (C# 2.0)	100
14 Struct Types	102
14.1 Boxing of Struct Values	104
14.2 The this Reference in a Struct	104
14.3 Struct Expressions: Value or Variable	104
15 Interfaces	106
15.1 Interface Declarations	106
15.2 Classes and Struct Types Implementing Interfaces	108
15.3 Explicit Interface Member Implementations	108
16 Enum Types	110
17 Delegate Types	112
18 Nullable Types over Value Types (C# 2.0)	114
19 Exceptions	116
20 Threads, Concurrent Execution, and Synchronization	118
20.1 Threads and Concurrent Execution	118
20.2 Locks and the lock Statement	120
20.3 Operations on Threads	122
20.4 Operations on Locked Objects	122
21 Mathematical Functions	124
22 Input and Output	126
22.1 Creating Streams	126
22.2 Overview of Input and Output Methods	128
22.3 Using Declarations, Exceptions, Thread Safety	128
22.4 Sequential Character Input: TextReaders	130
22.5 Sequential Character Output: TextWriters	132
22.6 Binary Input and Output: BinaryReader and BinaryWriter	134
22.7 Byte Input and Output: Stream	136
22.8 Directories	140
22.9 Files	140
22.10 Network Communication	142

23 Generic Types and Methods (C# 2.0)	144
23.1 Generics: Safety, Generality and Efficiency	144
23.2 Generic Types, Type Parameters, and Constructed Types	144
23.3 Generic Classes	146
23.4 Constraints on Type Parameters	148
23.5 Generic Interfaces	150
23.6 How Can Type Parameters Be Used?	150
23.7 Generic Methods	152
23.8 Generic Delegate Types	154
23.9 Abbreviations for Constructed Types	154
23.10 Generic Struct Types	156
23.11 The Implementation of Generic Types and Methods	156
24 Generic Collections: Lists and Dictionaries (C# 2.0)	158
24.1 The ICollection<T> Interface	158
24.2 Enumerators and Enumerables	160
24.3 Comparables, Equatables, Comparers and EqualityComparers	162
24.4 The IList<T> Interface	164
24.5 The IDictionary<K,V> Interface	164
24.6 The List<T> Class	166
24.7 The Dictionary<K,V> Class	168
24.8 The KeyValuePair<K,V> Struct Type	168
24.9 The SortedDictionary<K,V> and SortedList<K,V> Classes	168
24.10 The Queue<T> Class	170
24.11 The Stack<T> Class	170
25 Namespaces	172
25.1 The using Directive	172
26 Partial Type Declarations (C# 2.0)	174
27 Assertions and the Debug.Assert Method	176
28 Attributes	178
28.1 Some Predefined Attributes	178
28.2 Declaring and Using Custom Attributes	178
29 Main Differences Between C# and Java	180
30 References	182
Index	183