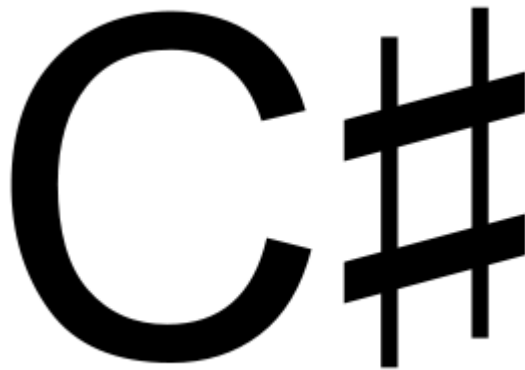


C# PROGRAMMING

by Wikibooks contributors



Developed on **Wikibooks**,
the open-content textbooks collection

© Copyright 2004–2007, Wikibooks contributors.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Principal authors: [Rod A. Smith](#) (C) · [Jonas Nordlund](#) (C) · [Jlenthe](#) (C) · [Nercury](#) (C) · [Ripper234](#) (C)

Cover: C# musical note, by [Mothmolevna](#) (*See naming*) (GFDL)

The current version of this Wikibook may be found at:
http://en.wikibooks.org/wiki/C_Sharp_Programming

Contents

INTRODUCTION.....	04
Foreword.....	04
Getting Started.....	06
LANGUAGE BASICS.....	08
Syntax.....	08
Variables.....	11
Operators.....	17
Data Structures.....	23
Control.....	25
Exceptions.....	31
CLASSES.....	33
Namespaces.....	33
Classes.....	35
Encapsulation.....	40
THE .NET FRAMEWORK.....	42
.NET Framework Overview.....	42
Console Programming.....	44
Windows Forms.....	46
ADVANCED OBJECT-ORIENTATION CONCEPTS.....	47
Inheritance.....	47
Interfaces.....	49
Delegates and Events.....	51
Abstract Classes.....	54
Partial Classes.....	55
Generics.....	56
Object Lifetime.....	59
ABOUT THE BOOK.....	61
History & Document Notes.....	61
Authors.....	62
GNU Free Documentation License.....	63

1 FOREWORD

[live version](#) · [discussion](#) · [edit chapter](#) · [comment](#) · [report an error](#)

C# (pronounced "See Sharp") is a multi-purpose computer **programming language** suitable for all development needs.

Introduction

Although C# is derived from the **C programming language**, it has features such as **garbage collection** that allow beginners to become proficient in C# more quickly than in **C** or **C++**. Similar to **Java**, it is **object-oriented**, comes with an extensive *class library*, and supports exception handling, multiple types of **polymorphism**, and separation of interfaces from implementations. Those features, combined with its powerful development tools, multi-platform support, and *generics*, make C# a good choice for many types of software development projects: **rapid application development** projects, projects implemented by individuals or large or small teams, Internet applications, and projects with strict reliability requirements. Testing frameworks such as **NUnit** make C# amenable to **test-driven development** and thus a good language for use with **Extreme Programming** (XP). Its **strong typing** helps to prevent many programming errors that are common in weakly typed languages.

A large part of the power of C# (as with other .NET languages), comes with the common .NET Framework API, which provides a large set of classes, including ones for encryption, TCP/IP socket programming, and graphics. Developers can thus write part of an application in C# and another part in another .NET language (e.g. VB .NET), keeping the tools, library, and object-oriented development model while only having to learn the new language syntax.

Because of the similarities between C# and the C family of languages, as well as **Java**, a developer with a background in object-oriented languages like C++ may find C# structure and syntax intuitive.

Standard

Microsoft, **Anders Hejlsberg** as Chief Engineer, created C# as part of their .NET initiative and subsequently opened its **specification** via the **ECMA**. Thus, the language is open to implementation by other parties. Other implementations include **Mono** and **DotGNU**.

C# and other .NET languages rely on an implementation of the **virtual machine** specified in the **Common Language Infrastructure**, like Microsoft's *Common Language Runtime* (CLR). That virtual machine manages memory, handles object references, and performs Just-In-Time (JIT) compiling of **Common Intermediate Language** code. The virtual machine makes C# programs safer

than those that must manage their own memory and is one of the reasons .NET language code is referred to as *managed code*. More like Java than C and C++, C# discourages explicit use of pointers, which could otherwise allow software bugs to corrupt system memory and force the operating system to halt the program forcibly with nondescript error messages.

History

Microsoft's original plan was to create a rival to Java, named J++ but this was abandoned to create C#, codenamed "Cool".

Microsoft submitted C# to the ECMA standards group mid-2000.

C# 2.0 was released in late-2005 as part of Microsoft's development suite, Visual Studio 2005. The 2.0 version of C# includes such new features as generics, partial classes, and iterators.

Se [microsoft-watch](#) and [hitmil](#).

[live version](#) · [discussion](#) · [edit chapter](#) · [comment](#) · [report an error](#)

[Click here to download full PDF material](#)