C# 10.0 All-in-One For Dummies Cheat Sheet

C# provides you with access to a phenomenal array of programming options. Use this cheat sheet to help you get the job done faster and easier when using C# as your development solution of choice.

A quick overview of C# 10.0 desktop application templates

One of the areas where Microsoft has made big changes in C# 10.0 is in the ability to run applications just about anywhere given the proper pre-requisites. Consequently, it's important for a developer to know which templates are available and which platforms they support.

The following table makes it easier for you to find a particular template for desktop development. Note that this list assumes that you're working with C# and not some of the other languages that Visual Studio 2022 supports.

Project Type	Supported Environments	Description
Blank App (Universal Windows)	Windows, Xbox, UWP, Desktop	Allows creation of applications using the new Universal Windows Platform (UWP) approach. The template doesn't provide any underlying application files.
Empty Project (.NET Framework)	Windows, Desktop	Allows creation of Windows Forms applications. The template doesn't provide any underlying application files. Note that this template uses older versions of the .NET Framework.
NUnit Test Project	Linux, macOS, Windows, Desktop, Web	Provides the means to test your applications no matter where they might reside. The same testing process works on both the Web and on the Desktop.
Shared Project	Windows, Desktop	Allows the creation of code that spans multiple projects. The goal is to allow building of the code differently based on the project in which it resides. You can learn more about this template type at https://docs.microsoft.com/visualstudio/extensibility/migration/update-visual-studio-extension.
Windows Application Packaging Project	Windows, UWP, Desktop	Creates an application package that is contained in a ~~.msix~~ file. The goal of this project is to create an application you can distribute through the Microsoft Store.
Windows Forms App	Windows, Desktop	Creates a traditional Windows Forms application for use in the desktop environment. This template lets you use the newer .NET Framework 5.0 and above features.
Windows Forms App (.NET Framework)	Windows, Desktop	Creates a traditional Windows Forms application for use in the desktop environment. Note that this project uses older versions of the .NET Framework.
Windows Forms Class Library	Windows, Desktop, Library	Allows development of reusable code modules for Windows Forms applications.
Windows Forms Control Library	Windows, Desktop, Library	Makes it possible to create custom or user controls for use in Windows Forms applications. This template lets you use the newer .NET Framework 5.0 and above features.
Windows Forms Control Library (.NET Framework)	Windows, Desktop, Library	Makes it possible to create custom or user controls for use in Windows Forms applications. Note that this template uses older versions of the .NET Framework.

Windows Service (.NET Framework)	Windows, Desktop, Service	Provides the means for creating an application that runs in the background as a service. Service setups are accessed through the Services MMC, rather than directly through a user interface. Note that this template uses older versions of the .NET Framework.
WPF App (.NET Framework)	Windows, Desktop	Uses the newer Windows Presentation Foundation (WPF) approach to creating applications. Note that this template uses older versions of the .NET Framework.
WPF Application	Windows, Desktop	Uses the newer Windows Presentation Foundation (WPF) approach to creating applications. This template lets you use the newer .NET Framework 5.0 and above features.
WPF Browser App (.NET Framework)	Windows, Desktop	Allows creation of a browser-like application for the desktop using the WPF approach. Note that this template uses older versions of the .NET Framework.
WPF Class Library	Windows, Desktop, Library	Allows development of reusable code modules for WPF applications.
WPF Custom Control Library	Widows, Desktop, Library	Makes it possible to create custom controls for use in a WPF application. This template lets you use the newer .NET Framework 5.0 and above features.
WPF Custom Control Library (.NET Framework)	Windows, Desktop, Library	Makes it possible to create custom controls for use in a WPF application. Note that this template uses older versions of the .NET Framework.
WPF User Control Library	Windows, Desktop, Library	Makes it possible to create user controls for use in a WPF application. This template lets you use the newer .NET Framework 5.0 and above features.
WPF User Control Library (.NET Framework)	Windows, Desktop, Library	Makes it possible to create user controls for use in a WPF application. Note that this template uses older versions of the .NET Framework.

C# 10.0 keyword listing

All programming languages rely on *keywords* — words that are reserved and that you can't use for your personal needs. Knowing the keywords makes it possible for you to choose other words for your code. For example, you couldn't create a variable named "while" because it's a C# keyword. The following table contains the C# keywords.

abstract	as	base	bool	break
Byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	goto
if	implicit	in	int	interface
internal	is	lock	long	namespace
new	null	object	operator	out
override	params	private	protected	public
readonly	ref	return	sbyte	sealed
short	sizeof	stackalloc	static	string
struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked
unsafe	ushort	using	virtual	void
volatile	while			

Common Visual Studio 2022 keyboard shortcuts

Touch typists find that using keyboard shortcuts makes them even faster than normal. Of course, you need to know the keyboard shortcuts before you can use them. The following table contains the keyboard shortcuts used most often in Visual Studio 2022. You can find a complete list of keyboard shortcuts at https://docs.microsoft.com/visualstudio/ide/default-keyboard-shortcuts-in-visual-studio.

Shortcut	**Purpose**
Alt+Enter	Show the properties panel for a selected object
Alt+Shift+A	Add an existing item to the project
Ctrl+Alt+L	Show Solution Explorer
Ctrl+Alt+P	Attach the debugger to a process
Ctrl+B, Ctrl+T	Toggle code bookmark
Ctrl+F	Display the Find dialog
Ctrl+F5	Start a project without debugging
Ctrl+H	Display the Replace dialog
Ctrl+K, Ctrl+C	Comment a selected block of code
Ctrl+K, Ctrl+U	Un-comment a selected block of code
Ctrl+M, Ctrl+M	Expand or collapse a selected code fragment (toggle)
Ctrl+M, Ctrl+O	Collapse all code to definitions
Ctrl+M, Ctrl+P	Expand all definitions to code
Ctrl+N	Add a new file
Ctrl+S	Save file
Ctrl+Shift+A	Add a new item to the project
Ctrl+Shift+B	Build solution
Ctrl+Shift+N	Close the current project and start a new project
Ctrl+Shift+O	Close the current project and open an existing project
Ctrl+Shift+F10	During debugging, set execution to the line holding the cursor
Ctrl+Shift+Tab	Scroll backwards through open windows
Ctrl+Tab	Scroll forward through open windows
Ctrl+Y	Redo typing
Ctrl+Z	Undo typing
F5	Start a project in debug mode
F6	Build solution
F7	Show the code window
F9	During debugging, toggle breakpoint on the line holding the cursor
F12	In the code editor, jump to the selected object’s definition
Shift+Alt+C	Add a new class to the project
Shift+F6	Build project
Shift+F7	Show the designer window