

C# Cheat Sheet

Data Types		Type Conversion Methods	Naming Conventions	
bool	Boolean value	ToBoolean	Class	MyClass
byte	8-bit unsigned integer	ToByte	Method	MyMethod
char	16-bit Unicode character	ToChar	Local variable	myLocalVariable
decimal	128-bit precise decimal values with digits	DateTime	Private variable	_myPrivateVariable
		ToDecimal	Constant	MyConstant
double	64-bit double-precision floating point	ToDouble	Arrays	
float	32-bit single-precision floating point	ToSingle		
int	32-bit signed integer	ToInteger	int[] array = new int[] {1, 2, 3}	
long	64-bit signed integer	ToLong	int[] array = {1, 2, 3}	
object	Base type for all other types	ToObject	var array = new int[] {1, 2, 3}	
sbyte	8-bit signed integer	ToSByte	int[] array = new int[3]	
short	16-bit signed integer	ToShort		
string	String value	ToString		
uint	32-bit unsigned integer	ToUInt32		
ulong	64-bit unsigned integer	ToUInt64		
ushort	16-bit unsigned integer	ToUInt16		

C# Cheat Sheet

Statements		Classes		Access Modifiers	
if-else	if (true) {...} {{nl}} else {{nl}} else {...}	Class	public class Dog {...}	public	Accessible by any other code in the same assembly or another assembly that references it
switch	switch (var) {{{nl}}}case default: break; }	Inheritance	public class Dog: Pet {...}	private	Only accessible by code in the same class or struct
for	for (int i =1; i < 5; i++)	Constructor (no parameters)	public Dog () {...}	protected	Only accessible by code in the same class or struct, or in a derived class
foreach-in	foreach (int item in array)	Constructor (one parameter)	public Dog (string var) {...}	internal	Accessible by any code in the same assembly, but not from another assembly
while	while (true) {...}	Field	public string name	protected internal	Accessible by any code in the same assembly, or by any derived class in another assembly
do... while	do {...} {{nl}} while (true)	Static Class	public static class Dog {...}	Cannot have modifiers or parameters	
try-catch-finally	try {...} {{nl}} catch (Exception) {{nl}} catch {...} {{nl}}	Static Member	public static int = 1		
		Finalizer (destructor)	~Dog () {...}		

C# Cheat Sheet

Other Modifiers		Other Modifiers (cont)		Comparison Operators	
abstract	Indicates that a class is intended to be a base class of other classes	virtual	Declares a method or an accessor whose implementation can be changed by a derived class	<	Less than
async	Indicates that the modified method is an asynchronous expression, or an anonymous method	volatile	Indicates that a field can be modified by something such as the operating system hardware, or a concurrently executing thread	>	Greater than
const	Specifies that the value of the field or variable cannot be modified			<=	Less than or equal to
event	Declares an event			>=	Greater than or equal to
extern	Indicates that the method is implemented in another assembly			==	Equal to
new	Explicitly hides a member inherited from a base class			!=	Not equal to
override	Provides a new implementation of a method inherited from a base class	Assignment Operators			
partial	Defines partial classes, structs and interfaces throughout the same assembly	=	Simple assignment	Arithmetic Operators	
readonly	Declares a field that can only be assigned in the declaration or in a constructor of the same class	+=	Addition assignment	+	Add numbers
sealed	Specifies that a class cannot be inherited	-=	Subtraction assignment	-	Subtract numbers
static	Declares a member that belongs to the type instead of to a specific object	*=	Multiplication assignment	*	Multiply numbers
unsafe	Declares an unsafe context	/=	Division assignment	/	Divide numbers
		%=	Remainder assignment	%	Compute remainder of division of numbers
		&=	AND assignment	++	Increases integer value by 1
		=	OR assignment	--	Decreases integer value by 1
		^=	XOR assignment		
		<<=	Left-shift assignment		
		>>=	Right-shift assignment		

C# Cheat Sheet

Logical and Bitwise Operators	
&&	Logical AND
	Logical OR
!	Logical NOT
&	Binary AND
	Binary OR
^	Binary XOR
~	Binary Ones Complement
<<	Binary Left Shift
>>	Binary Right Shift

Other Operators	
sizeof()	Returns the size of a data type
typeof()	Returns the type of a class
&	Returns the address of a variable
*	Pointer to a variable
? :	Conditional expression
is	Determines whether an object is of a specific type
as	Cast without raising an exception if the cast fails