H1 Quick Reference	
Create a new console application using Visual Studio 2013	On the File menu, point to New, and then click Project to open the New Project dialog box. In the left pane, under Installed Templates, click Visual C#. In the middle pane, click Console Application. in the Location box, specify a directory for the project files. Type a name for the project and then click OK.
Create a new Windows Store blank graphical application for Windows 8.1 using Visual Studio 2013	On the File menu, point to New, and then click Project to open the New Project dialog box. In the left pane, in the Installed Templates section, expand Visual C#, and then click Windows Store. In the middle pane, click Blank App (XAML). In the Location box, specify a directory for the project files. Type a name for the project and then click OK.
Create a new WPF graphical application for Windows 7 or Windows 8 using Visual Studio 2013	On the File menu, point to New, and then click Project to open the New Project dialog box. In the left pane, in the Installed Templates section, expand Visual C#, and then click Windows. In the middle pane, click WPF Application. Specify a directory for the project files in the Location box. Type a name for the project and then click OK.
Build the application	On the Build menu, click Build Solution.
Run the application in Debug mode	On the Debug menu, click Start Debugging.
Run the application without debugging	On the Debug menu, click Start Without Debugging.

H2 Quick Reference	
Declare a variable	Write the name of the data type, followed by the name of the variable, followed by a semicolonFor example: int outcome;
Declare a variable and give it an initial value	Write the name of the date type, followed by the name of the variable, followed by the assignment operator and the initial value. Finish with a semicolon. For example: int outcome = 99;
Change the value of a variable	Write the name of the variable on the left, followed by the assignment operator, followed by the expression calculating the new value, followed by a semicolon. For example: outcome = 42;
Generate a string representation of the value in a variable	Call the ToString method of the variable. For example: int intVar = 42; string stringVar = intVar.ToString();
Convert a string to an int	Call the System.Int32.Parse method. For example: string stringVar = "42"; int intVar = System.Int32.Parse(stringVar);
Override the precedence of an operator	Use parentheses in the expression to force the order of evaluation. For example: (3 + 4) * 5
Assign the same value to several variables	Use an assignment statement that lists all the variables.For example: myInt4 = myInt3 = myInt2 = myInt = 10;
Increment or decrement a variable	Use the ++ or -- operator. For example: count++;

Index	
statement	a command that performs an action.
syntax	well-defind set of rules, describing format and construction
semantics	specification of what things do
identifier	a name to identify an element
variable	a storage location to hold a value
oparator	operate on operands (values) to create new values
operands	the values on wich an operator performs its function

Index (cont)

NaN	Not a Number
precedence	govern's the order in wich an expression's operators are evaluated.
associativity	the direction in wich the operands of an operator are evaluated.
method	a named sequence of statements

Keywords

reserved keywords	catch false namespace short ushort char finally new sizeof using checked fixed null stackalloc virtual class float object static void const for operator string volatile continue foreach out struct while decimal goto override switch default if params this delegate implicit private throw
not to be used keywords	add get remove alias global select ascending group set async into value await join var descending let where dynamic orderby yield from partial

Naming conventions

Important	C# is case-sensitive !
identifiers	only letters (upper and lowercase), digits and underscore. Must start with letter or underscore
variables	Use camelcase (as camelCase). don't start with underscore. Don't differ only by case. Never use Hungarian notation.

Primitive Data Types

Data type	Description	Size (bits)	Range	Sample usage
int	Whole numbers (integers)	32	-2^{31} through $2^{31}-1$	int count; count = 42;
long	Whole numbers (bigger range)	64	-2^{63} through $2^{63}-1$	long wait; wait = 42L;
float	Floating-point numbers	32	$\pm1.5 \times 10^{-45}$ through $\pm3.4 \times 10^{38}$	float away; away = 0.42F;
double	Double-precision (more accurate) floating-point numbers	64	$\pm5.0 \times 10^{-324}$ through $\pm1.7 \times 10^{308}$	double trouble; trouble = 0.42;
decimal	Monetary values	128	28 significant figures	decimal coin; coin = 0.42M;
string	Sequence of characters	16 bits per character	Not applicable	string vest; vest = "forty two";
char	Single character	16	0 through $2^{16}-1$	char grill; grill = 'x';
bool	Boolean	8	True or false	bool teeth; teeth = false;

H3 Quick Reference	
Declare a method	Write the method within a class. Specify the method name, parameter list, and return type, followed by the body of the method between braces. For example: int addValues(int leftHandSide, int rightHandSide) { ... }
Return a value from within a method	Write a return statement within the method. For example: return leftHandSide + rightHandSide;
Return from a method before the end of the method	Write a return statement within the method. For example: return;
Call a method	Write the name of the method together with any arguments between parentheses. For example: addValues(39, 3);
Use the Generate Method Stub Wizard	Right-click a call to the method and then, on the shortcut menu, click Generate Method Stub.
Display the Debug toolbar	On the View menu, point to Toolbars, and then click Debug.
Step into a method	On the Debug toolbar, click Step Into. or On the Debug menu, click Step Into.
Step out of a method	On the Debug toolbar, click Step Out. or On the Debug menu, click Step Out.
Specify an optional parameter to a method	Provide a default value for the parameter in the method declaration. For example: void optMethod(int first, double second = 0.0, string third = “Hello”) { ... }
Pass a method argument as a named parameter	Specify the name of the parameter in the method call. For example: optMethod(first : 100, third : “World”);
mixed positional and named arguments	First specify all positional arguments, then named arguments.