

# Core C# and .NET

## Quick Reference

### 1. Data Types

Primitive	Size	Example
<b>string</b>	2 bytes/char	s = "reference";
<b>bool</b>		b = true;
<b>char</b>	2 bytes	ch = 'a';
<b>byte</b>	1 byte	b = 0x78;
<b>short</b>	2 bytes	ival = 54;
<b>int</b>	4 bytes	ival = 540;
<b>long</b>	8 bytes	ival = 5400;
<b>float</b>	4 bytes	val = 54.0F;
<b>double</b>	8 bytes	val = 54.0D;
<b>decimal</b>	16 bytes	val = 54.0M;

### 2. Arrays

#### Declaration

```
int[] numArray = {1903, 1907, 1910}; int[] numArray = new
int[3];
// 3 rows and 2 columns
int[, ] nums = {{1907, 1990}, {1904, 1986}, {1910, 1980}};
```

#### Array Operations

```
Array.Sort(numArray); // sort ascending
// Sort begins at element 4 and sorts 10 elements
Array.Sort(numArray, 4, 10);
// Use one array as a key and sort two arrays
string[] values = {"Cary", "Gary", "Barbara"}; string[] keys = {"Grant",
"Cooper", "Stanwyck"}; Array.Sort(keys, values);
// Clear elements in array (array, 1st element, # elements)
Array.Clear(numArray, 0, numArray.Length);
// Copy elements from one array to another
Array.Copy(src, target, numElements);
```

### 3. String Operations

Method	Description
<b>Compare</b>	String.Compare(stra, strb, case, ci) bool case – true for case insensitive ci – new CultureInfo("en-US") returns: <0 if a<b, 0 if a=b, 1 if a>b
<b>IndexOf</b>	str.IndexOf(val, start, num) val – string to search for start – where to begin in string num – number of chars to search returns (-1) if no match.
<b>LastIndexOf</b>	Search from end of string.
<b>Replace</b>	new str = oldstr.Replace("old", "new");
<b>Split</b>	Char[] delim = { ' ', ',' }; string w = "Kim, Joanna Leslie"; // create array with three names string[] names = w.Split(delim);

Method	Description
<b>Substring</b>	mystring.Substring(ndx, len) string alpha = "abcdef"; // returns "cdef" string s = alpha.Substring(2); // returns "de" s = alpha.Substring(3, 2);
<b>ToCharArray</b>	Places selected characters in a string in a char array:  String vowel = "aeiou"; // create array of 5 vowels char[] c = vowel.ToCharArray(); // create array of 'i' and 'o'. char[] c = vowel.ToCharArray(2, 2);

### 4. System.Text.StringBuilder

#### Constructor

```
StringBuilder sb = new StringBuilder(); StringBuilder sb = new
StringBuilder(mystring);
StringBuilder sb = new StringBuilder(mystring, capacity);
```

*mystring* – Initial value of StringBuilder object *capacity* –  
Initial size (characters) of buffer.

```
Using StringBuilderMembers decimal bmi =
22.2M;
int wt = 168;
StringBuilder sb = new StringBuilder("My weight is "); sb =
sb.Append(wt); // can append number
sb = sb.Append(" and my bmi is ").Append(bmi);
// my weight is 168 and my bmi is 22.2 sb =
sb.Replace("22.2", "22.4");
string s = sb.ToString();
// Clear and set to new value
sb.Length = 0; sb.Append("Xanadu");
```

### 5. DateTime and TimeSpan

#### DateTime Constructor

```
DateTime(yr, mo, day)
DateTime(yr, mo, day, hr, min, sec)
```

```
DateTime bday = new DateTime(1964, 12, 20, 11, 2, 0); DateTime
newyr = DateTime.Parse("1/1/2005"); DateTime currdt =
DateTime.Now;
// also AddHours, AddMonths, AddYears DateTime
tomorrow = currdt.AddDays(1); TimeSpan diff =
currdt.Subtract(bday);
// 14795 days from 12/20/64 to 6/24/05
Console.WriteLine("{0}", diff.Days);
```

```
// TimeSpan(hrs, min, sec)
TimeSpan ts = new TimeSpan(6, 30, 10);
// also FromMinutes, FromHours, FromDays TimeSpan ts =
TimeSpan.FromSeconds(120); TimeSpan ts = ts2 - ts1; // +, -,
>, <, ==, !=
```

### 6. C# Language Fundamentals

Control Flow Statements	
<b>switch</b> (expression) { case expression: // statements break / goto / return() case ... default: // statements break / goto / return() } <i>expression</i> may be integer, string, or enum.	sw itch (genre) { case "vhs": price = 10.00M; break; case "dvd": price = 16.00M; break; default: price = 12.00M; break; }
<b>if</b> (condition) { // statements } else { // statements }	if (genre == "vhs") price = 10.00M; else if (genre == "dvd") price = 16.00M; else price = 12.00M;
Loop Constructs	
<b>while</b> (condition) { body }	w hile ( ct < 8) { tot += ct; ct++; }
<b>do</b> { body } w hile (condition);	do { tot += ct; ct++; } w hile (ct < 8);
<b>for</b> (initializer; termination condition; iteration;) { // statements }	for (int i=0; i<8; i++) { tot += i; }
<b>foreach</b> (type identifier in collection) { // statements }	int[] ages = {27, 33, 44}; foreach (int age in ages) { tot += age; }