

C# Programming Reference Sheet

Built In Data Types & Literals

Integers

int, byte, short, long

Floating Point Numbers

double, decimal, float

Strings and Characters

String, char

Boolean

bool (true or false)

Simple Programming Statements

Constant declaration

const int;

Variable declaration

string message;

Assignment

message = "Welcome";

Method call

Console.ReadLine();

Sequence of statements - grouped

```
{  
    }  
}
```

Declaring Methods

Declare a method with parameters:

```
public static void addstr(string p1,  
string p2, string p3){ }
```

Declare a method that returns data:

```
public static int myMethod() { }  
//return int value 7 return 3+4;
```

Pass by reference:

```
public void myMethod(ref int x) {x = 10;}  
(The passed parameter can be modified by the  
method so the value passed is the memory location of  
the argument.)
```

Custom Types

Classes

```
class Bike  
{  
    public string color = "white";  
}
```

Enumerations

```
enum Color { Black, White, Brown }
```

Structs

```
struct_body  
: '{ struct_member_declaration* }'  
;
```

Programs and Modules

Creating a program

```
namespace program {  
    class mainprogram {  
    }  
}
```

Using a class from a library

```
#include Swin;
```

Working with Strings

Assignment (giving a string a value)

```
string x = "Welcome";
```

Concatenation (joining strings)

```
string y = x + " Mate!";
```

Comparison

```
x == y, x == "Welcome", "Welcome" != "Mate"
```

Construction from other types:

```
string x = a.ToString();  
string y = Convert.ToString(4);
```

Structured Programming Statements

If statement

```
If( condition ){ then } else { then }
```

Case statement

```
switch( variable ) {default: break;}
```

While loop

```
while (condition) {do}
```

Repeat loop

```
do { } while (condition)
```

For loop

```
for(int i =0;i < 10;i++) { }
```

Boolean Operators and Other Statements

Comparison: equal, less, larger, not equal, less eq

```
==, , !=, <=
```

Boolean: And, Or and Not

```
&&, ||, !
```

Skip an iteration of a loop

```
continue;
```

End a loop early

```
break;
```

End a method:

```
return;
```

Arrays

Declaration

```
String[ ] bikes = new String[10];  
Rider[ ] riders = new Rider[4];
```

Access

```
bikes[0]; riders[10];
```

Loop with index i

```
for(int i=0;i< bike.length; i++) { }
```

For each loop

```
foreach (string bike in bikess) { }  
foreach (Rider rider in riders) { }
```

Other Things

Reading from Terminal

```
Console.ReadLine();
```

Writing to Terminal

```
Console.WriteLine(" ");  
Console.Write("");
```

Comments

```
//  
/*  
onetwothreefourfive  
*/
```