

C# Programming Reference Sheet

Built In Data Types & Literals

Integers

Int, Short, Long
(eg: 5, 10, 15)

Floating Point Numbers

decimal, Double, Float
(eg: 3.1, 2.5, 2.1)

Strings and Characters

String, Char (eg: 'Hello', 'H')

Boolean

Bool (eg: True, False)

Simple Programming Statements

Constant declaration

```
const data_type constant_name = value;
```

Variable declaration

```
Data_type varname;
```

Assignment

```
String name = 'Fred'; int age = 5;
```

Procedure Call

```
Sort(p1, p1);
```

Sequence of statements - grouped

```
{ content }
```

Declaring Functions & Procedures

Declare a procedure with parameters:

```
Public Void Sort(int v1, v2) {}
```

Declare a functions:

```
Public int print() {}
```

Pass by reference:

```
Public void Swap (ref int r1){}  
Public void swap (int x, int y) {}  
Public void swap (out int x) {}
```

Working with Strings

Assignment (giving a string a value)

```
String name = "Fred";
```

Concatenation (joining strings)

```
String name = name + " Smith";
```

Comparison

```
if (name == "Fred Smith") {}
```

Construction from other types:

```
varname.ToString();  
(int)notintvalue
```

Structured Programming Statements

If statement

```
if (condition) {statements}
```

case statement

```
int value = 5; switch(value)  
{case 1:statement; break;}
```

while loop

```
while(conditions) {statements}
```

repeat loop

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

For loop

```
for (int i = 0; a < 20; a++;
```

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 function/procedure:

```
Return result; Return;
```

Custom Types

Records/Class

```
Class player {public double length;  
public string name;}
```

```
Declaring: player player1 = new player();
```

```
Assigning: player1.name = "name";
```

Enumerations

```
Enum Days {Monday,Tuesday, etc};  
Days CurDay = Monday;
```

Arrays

Declaration

```
datatype[] arrayName;
```

Access

```
scores[0] = 10;
```

Loop

```
for ( i = 0; i < 10; i++ ){arrayname[i] = i+ 100;}
```

use a foreach statement to iterate through an array. `Foreach(string value in arrayName) {statem}`

Programs and Modules

Creating a program

```
Namespace Nameof Program  
{  
Content  
}
```

Using a module

```
using System;  
namespace asdf{}
```

Other Things

Reading from Terminal

```
Console.Read() Console.ReadLine()
```

Writing to Terminal

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

Comments

```
/// single line accepts xml  
// text comments  
/* multi line comments*/
```

Compiling

In engine visual studio