C# Programming Reference Sheet

Built In Data Types & Literals

Integers:
Short, Int, long (-5000, 50, 99999999999)
Floating Point Numbers:
Float, Double (5.5, 1.21)
Strings and Characters:
String, Char (“ABCD”, “A”, “hello”)
Boolean:
Bool (true, false , yes , no)

Simple Programming Statements

Constant declaration:
public const string greeting = “hello”;
Variable declaration
string name;
Assignment
X = 1;
Method call
Draw();
Sequence of statements - grouped: between curly brackets{ … }

Declaring Methods

Declare a method with parameters:
<access><returntype><methodname>(params){}
Public int AddNumbers (int num1, int num2){}
Declare a method that returns data:
Public int AddNumbers (int num1, int num2){}
{Return num1 + num2;}
Pass by reference:
Keyword: ref
Int num =1;
Method (ref num);
Console.WriteLine(num);

Custom Types

Classes (custom data type)
Public class Books { }
Enumerations
enum Genre {rock,classic,pop,metal,dance}
Structs
struct student {public string name; public int scores;}

Programs and Modules

Creating a program

```
class Program
{
   static void Main(string[] args)
    {
    }
 }
```

Using a class from a library

```
using SplashKitSDK;
       SplashKit.ProcessEvents();
```

Working with Strings

Assignment: (giving a string a value)
String name = “conrad”;
Concatenation (joining strings)
String example = “John” + “smith”;
Comparison : ==
If number == number2;
Construction from other types:
int num = 5;
String msg = num.ToString();

Structured Programming Statements

If statement:
if (num1 < num2)..{} else
Case statement: switch
Switch(x)case 1 :…;break; case 2 :…;break
default:…;break
While loop:
while (condition){ //code}
Repeat loop: runs at least once then repeats - do, while:
do { //code} while (condition)
For loop:
for (int I =0; I < count; 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
Int[] numbers = {…};
Access
Number[0] = 50;
Loop with index i
for (i=0;i<5;i++) {number[i] =i;}
For each loop
Foreach (int i in cars) {Console.WriteLine(i)}

Other Things

Reading from Terminal
Console.ReadLine()
Writing to Terminal
Console.WriteLine()
Comments
// - single line
/* */ - whole block of code