C# Programming Reference Sheet

Built In Data Types & Literals

Integers
```
Int, short, long (eg: 5, 10)
```
Floating Point Numbers
```
Float, double (eg: 3.1, 2.5)
```
Strings and Characters
```
String, char (eg:"qwerty", "a")
```
Boolean
```
Bool (eg: True, False)
```

Working with Strings

Assignment (giving a string a value)
```
Name = "joe";
```
Concatenation (joining strings)
```
Name = Name + " Joe";
```
Comparison
```
If(Name == "joe Joe") {}
```
Construction from other types:
```
Name = IntValue.ToString();
```

Simple Programming Statements

Constant declaration
```
Const int X;
```
Variable declaration
```
Public int i;
```
Assignment
```
i = 0;
```
Method call
```
Console.WriteLine("asdfgh");
```
Sequence of statements - grouped
```
{ ///////code }
```

Structured Programming Statements

If statement
```
If(Name == "joe Joe") {}
```
Case statement
```
Switch(intValue)
{case 1: Console.WriteLine("1"); break;}
```
While loop
```
While(I < 9000) {}
```
Repeat loop
```
Do{ ///code } While(i < 9000)
```
For loop
```
For(int i = 0; i < 5; i++) {}
```

Declaring Methods

Declare a method with parameters:
```
Public void CallNo(int number){}
```
Declare a method that returns data:
```
Public int CallNo (){return number;}
```
Pass by reference:
```
Int I;
CallNo(ref I);
Static void CallNo(ref int 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 value; Return;
```

Custom Types

Classes
```
Public class Customer{}
```
Enumerations
```
Public enum day{Monday,Tuesday}
```
Structs
```
Public struct studentID
{
      Public string name;
      Public int ID;
}
```

Arrays

Declaration
```
Int[] I;
Int[] I = new Int[3];
```
Access
```
I[0] = 5;
```
Loop with index i
```
For(int i = 0; i < 5; i++)
{Number[i]++}
```
For each loop
```
Foreach(int I in Iarray) {}
```

Programs and Modules

Creating a program
```
Namespace programName
{ class MainClass{}}
```
Using a class from a library
```
Using System
Using SwinGame;
```

Other Things

Reading from Terminal
```
Console.ReadLine();
```
Writing to Terminal
```
Console.WriteLine("Hello World!");
```
Comments
```
// single line
/* multiple line */
```