C# Programming Reference Sheet

Built in Data Types & Literals

Integers
short, int, long, eg: -5, 20, 100

Floating Point Numbers
float, double, decimal, eg: 4.2, 8.47, 85.65

Strings and Characters
string, char, eg: "Hello", "H"

Boolean
bool, eg: True, False

Working with Strings

Assignment (giving a string a value)
str = "String"

Concatenation (joining strings)
"Hello" + "World"

Comparison
if "String" == "String"

Construction from other types:
str = String.Concat("Hello", 11)

Simple Programming Statements

Constant declaration
eg: const, public const

Variable declaration
eg: int Number, byte Checksum

Assignment
Number = 0, Checksum = 0x00

Method call
Console.WriteLine(number, checksum)

Sequence of statements – grouped
begin … end;

Structured Programming Statements

If statement
eg: If (statement) {} … else if {} … else{}

Case statement
eg: switch(var) … case n: statement; break;

While loop
eg: while(condition){}

Repeat loop
eg: do{} while(done condition)

For loop
eg: for(counter, condition, increment) {}

Declaring Methods

Declare a method with parameters:
Static void methodName(){}

Declare a method that returns data:
static int methodName(){}

Pass by reference:
static int methodName(int y, int x){}

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

Custom Types

Classes
class className{}

Enumerations
enum enumName{}

Structs
struct structName {}

Arrays

Declaration
Byte[] arrayName, int[] arrayName[2] = {0,1,2}

Access
arrayName[2] = 2

Loop with index i
for(int I = 0, condition, i++) {}

For each loop
foreach (int number in numbers) {}

Programs and Modules

Creating a program

```
Class Program{
   Static void Main(string[] args){
      // program here
}}
```

Using a class from a library
using System, using UtilityLibraries

Other Things

Reading from Terminal
Console.ReadLine()

Writing to Terminal
Console.WriteLine("Output Here")

Comments
// (single line, / ….. */ (multi line*