# C# Programming Reference Sheet

Built In Data Types & Literals

Integers

Int, short, long

(e.g. 5, 10, 15)

Floating Point Numbers

Float, double  
(e.g. 3.14, 2.5555555, 69.69)

Strings and Characters

String char (e.g. “hello”, ‘h’)

Boolean  
Bool(e.g. true, false)

Working with Strings

Assignment (giving a string a value)

String Name = “Gamaliel”

Concatenation (joining strings) string name = “Gamaliel ” + “D’mello”

Comparison

String.Compare(str1, str2)

Construction from other types:

String text = age.ToString()

Structured Programming Statements

If statement

If (comparison using ==, =>, <=, < or >)

Case statement

switch (a)

{

Case 1: //code

Break;

Case 2: //code 2  
 break;

Default: //default if neither 1 or 2

Break;

}

While loop

While (i < 10) {do these steps}

Repeat loop repeat … until

do

{

//code

} while (condition);

For loop

for{i=0 ; i<count ; i++}

Simple Programming Statements

Constant declaration: Public Const **float** PI = 3.14f

Variable declaration: var var\_name

Assignment: PI = 3.14

Method call: Console.WriteLine(“text”)

Sequence of statements – grouped

If …. else

Declaring Methods

Declare a method with parameters:

Public void Print(“text passed as arg”)

{…}

Declare a method that returns data:

Public int addtwo(var One, var Two)

Return result = One + Two

Pass by reference:

Public int swap(var One, var Two)

{

…

}

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;

Other Things

Reading from Terminal: Console.Read

Writing to Terminal: Console.Write

Comments: // (single line) or /\* \*/ (multiline)

Programs and Modules

Creating a program

using System;

namespace HelloWorld

{class Hello {static void Main() {

Console.WriteLine("Hello World!");}}}

Using a class from a library

Using System;

Arrays

Declaration  
int die[] = new int[5]

Access: die[2], die[3]

Loop with index i

While i<die.length

Die[i]; i++

For each loop

For each (roll r in die){do this}

Custom Types

Classes

Public class x{ }

Enumerations: enum week{Monday, Tuesday, ETC. }

Structs: struct Employee{

public int EmpId;

public string FirstName;

public string LastName;

}