# C# Programming Reference Sheet

Built In Data Types & Literals

Integers

Short, Int, long (signed types)

(-1,3, 100)

Floating Point Numbers

float, double, (1.2, -3.14)

Strings and Characters

string, char, (”hello”, ‘h’…)

Boolean

bool

Working with Strings

Assignment (giving a string a value)

string name = “Jordan”

Concatenation (joining strings)

Concatenated = name + “extra”;

Comparison

If(name == “Jordan”){ }

Construction from other types:

int iNum = 2;

string sNum = iNum.ToString();

Structured Programming Statements

If statement

if(bool){}

Case statement

Switch(condition){

Case 1:…; break; Default:…; break;

While loop

while(bool){}

Repeat loop (do?)

do{}while(bool)

For loop

for(int i = 0; i < 10; i++){}

Programs and Modules

Creating a program

Using a class from a library

Custom Types

Classes

class foo{

public foo(args){}

…}

Enumerations

enum Height{Short, Average, Tall}…

Height enumVar = Height.Average;

Structs

Arrays

Declaration

int[] numArray = new int[size];

Access

int num = numArray[idx];

Loop with index i

for(int i = 0; i < numArray.Length; i++){}

For each loop

Foreach(int i in numArray){}

Other Things

Reading from Terminal

string input = Console.ReadLine();

Writing to Terminal

Console.WriteLine(“Hi!”);

Comments

//Not parsed

Declaring Methods

Declare a method with parameters:

void Method(type arg, ..){}

Declare a method that returns data:

int Method(){return 42;}

Pass by reference:

void Method(ref int arg){arg++;}

int num = 10;

Method(ref num);

Console.WriteLine(num); //num == 11

Simple Programming Statements

Constant declaration

const int GRAVITY = 9.81;

Variable declaration

int age; string name;

Assignment

age = 2; name = “Jordan”;

Method call

Console.WriteLine(string);

Sequence of statements - grouped

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;