# C# Programming Reference Sheet

Built In Data Types & Literals

Integers

Integer *variable*;

Floating Point Numbers

Float *variable*;

Strings and Characters

String *variable*;

Char *variable*;

Boolean

Boolean *variable*;

Working with Strings

Assignment (giving a string a value)

string = new string(value);

Concatenation (joining strings)

string = string1 + string2;

Comparison

If (string == string1) {}

Construction from other types:

String = Console.ReadLine(string1 + “ “ + string3[2]);

Simple Programming Statements

Constant declaration

Const *variable;*

Variable declaration

*varType* *varName*;

Assignment

*varName* = *result*;

*varName* = new *varType* *result*;

Method call

*methodName*();

*methodName*(*variable*);

Sequence of statements – grouped

Iteration: Top -> Bottom

Structured Programming Statements

If statement

if (*condition*)

{

//code

}

Case statement

switch(*variable*)

{

case *result*;

//code

break;

default:

//code

}

While loop

while(*condition*)

{

//code

}

Repeat loop

do

{

//code

} while(*condition*)

For loop

For (declaration, condition, iterator)

{

//code

}

Declaring Methods

Declare a method with parameters:

void methodName()

{

//code

return;

}

Declare a method that returns data:

void methodName(variable)

{

//code

return variable;

}

Pass by reference:

Other Things

Reading from Terminal

Console.ReadLine();

Writing to Terminal

Console.WriteLine()

Comments

//This is a message!

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;

Programs and Modules

Creating a program

Using a class from a library

className objectName = new className();

Arrays

Declaration

dataType[] arrayName;

arrayName = new dataType[] {“text”,…};

Access

arrayName[index]

Loop with index i

foreach(int i in arrayName)

{

Console.WriteLine(arrayName[i]);

}

For each loop

foreach(int variable in arrayName)

{

//code

}

Custom Types

Classes

public className(string argument1)

{

variable1 = argument1;

}

Enumerations

enum enumName

{

One = 1;

Two = 2;

Three = 3;

}

Structs

struct structName

{

public int variable;

public string variable;

}

**Reference through:**

structName var = new structName()

🡪‘var.variable’

# References

* <https://www.programiz.com/c-programming/c-switch-case-statement>
* <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/equality-operators>
* <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/do>
* <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/boolean-logical-operators>
* <https://stackoverflow.com/questions/654113/how-do-i-skip-an-iteration-of-a-foreach-loop>
* <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/foreach-in>
* <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/classes>
* <https://www.tutorialsteacher.com/csharp/csharp-enum>
* https://www.tutorialsteacher.com/csharp/csharp-struct