C# Programming Reference Sheet

Built In Data Types & Literals

Integers:
Int – (2,7)

Floating Point Numbers:
Float, Double – (8.9, 56.991)

Strings and Characters:
string, char – (‘y’ , “you”)

Boolean: bool – (0,1)

Strings

Assignment (giving a string a value)
name = “Ekrar”

Concatenation (joining strings):
Name = “Mike” + “Kenny”

Comparison
if(name == “ekrar”) {//do stuff}
Conversion from other types: x=2
string y = “I want to eat” + x.ToString() + “ apples”

Simple Programming Statements

Constant declaration
private const double avogadro = 6.02 X 10^23

Variable declaration
string name;

Assignment
Year = 2032

Method call= age.ToMonths();

Sequence of statements = grouped: {}

Structured Programming Statements

If statement = if(i>20) {condition}

Case statement=
Switch (expression){case y : break;}

While loop= while (condition){block}

Repeat loop= do {//statement} while (condition)

For loop = for (int i = 6 ; i<=n; i++) {statement}

Declaring Methods

Declare a method with parameters: public void Print(string s) {….}

Declare a method that returns data: public int sum(int[] Nums){..... }

Pass by reference: int Avg (ref int refSum) {..}

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 [data type] ;

Custom Types

Classes = class Shapedrawer(//code)

Enumerations = enum months {Jan, Feb, Mar};

Structs= structs Car{
Public string make;
Public string model;
Public int year; }

Arrays

Declaration = String[] grade;

Access = Grade[1];

Loop with index I =
for (int i=0; i< cars.Length(); i++) {}

For each loop
Foreach (string car in cars) {//statement}

Programs and Modules

Creating a program
Namespace myProgram{
Class MyProgram{
Static void Main()(….)
}
}

Using a class from a library= using Shapedrawer;
Namespace myprogram{
}

Other Things

Reading from Terminal = Console.Read() / Console.ReadLine()

Writing to Terminal = Console.Write() / Console.WriteLine()

Comments
//Single Line comments
*/
Multi line comments */