Cheatography

See Sharp Cheat Sheet

by Kemmojoo via cheatography.com/28261/cs/8310/

Comments

```
// Single line
/* Multiple
line */
/// XML comments on
single line
/* XML comments on
multiple lines /
```

Enumerations

```
enum Action {Start,
Stop, Rewind, Forward};
enum Status {Flunk =
50, Pass = 70, Excel =
90};
Action a = Action.Stop;
if (a != Action.Start)
//Prints "Stop is 1"
  System.Console.WriteLi
ne(a + " is " + (int)
a);

// Prints 70
System.Console.WriteLin
e((int) Status.Pass);
// Prints Pass
System.Console.WriteLin
e(Status.Pass);
enum Weekdays{
  Saturday, Sunday,
Monday, Tuesday,
Wednesday, Thursday,
Friday
}
```

Loops

```
//Pre-test Loops: while
(i < 10)
  i++;
```

Loops (cont)

```
for (i = 2; i < = 10; i +=
2)
  System.Console.WriteLine(i
);
//Post-test Loop:
do
  i++;
while (i < 10);
// Array or collection
looping
string[] names =
{"Steven", "SuOk",
"Sarah"};
foreach (string s in names)
  System.Console.WriteLine(s
);
```

Namespaces

```
namespace
ASPAlliance.DotNet.Community
{
  ...
}

// or

namespace ASPAlliance {
  namespace DotNet {
    namespace Community {
      ...
    }
  }
}
using
ASPAlliance.DotNet.Community
;
```

Objects

```
TopAuthor author = new
TopAuthor();

//No "With" construct
author.Name = "Steven";
author.AuthorRanking = 3;

author.Rank("Scott");
TopAuthor.Demote() //Calling
static method

TopAuthor author2 = author
//Both refer to same object
author2.Name = "Joe";
System.Console.WriteLine(auth
or2.Name) //Prints Joe

author = null //Free the
object

if (author == null)
  author = new TopAuthor();

Object obj = new
TopAuthor();
if (obj is TopAuthor)
  SystConsole.WriteLine("Is
a TopAuthor object.");
```

Delegates / Events

```
delegate void
MsgArrivedEventHandler(string
message);

event MsgArrivedEventHandler
MsgArrivedEvent;

//Delegates must be used with
events in C#
```

Delegates / Events (cont)

```
MsgArrivedEvent += new
MsgArrivedEventHandler
  (My_MsgArrivedEventC
allback);
//Throws exception if
obj is null
MsgArrivedEvent("Test
message");
MsgArrivedEvent -= new
MsgArrivedEventHandler
  (My_MsgArrivedEventC
allback);

using
System.Windows.Forms;

Button MyButton = new
Button();
MyButton.Click += new
System.EventHandler(MyB
utton_Click);
private void
MyButton_Click(object
sender,
System.EventArgs e) {
  MessageBox.Show(this,
"Button was clicked",
"Info",
    MessageBoxButtons.O
K,
MessageBoxIcon.Informat
ion);
}
```

Cheatography

See Sharp Cheat Sheet

by Kemmojoo via cheatography.com/28261/cs/8310/

Program Structure

```
using System
Namespace MyNameSpace{
  class HelloWorld {
    static void
Main(string[] args) {
      System.Console.Write
Line("Hello World")
    }
  }
}
```

Operators

```
//Comparison
== < > <= >= !=

//Arithmetic
+ - * /
% (mod)
/ (integer division if
both operands are ints)
Math.Pow(x, y)

//Assignment
= += -= *= /= %= &= |= ^=
<<= >>= ++ --

//Bitwise
& | ^ ~ << >>

//Logical
&& || !

//String Concatenation
+
```

Functions

```
// Pass by value (in,
default), reference
//(in/out), and reference
(out)
void TestFunc(int x, ref
int y, out int z) {
  x++;
  y++;
  z = 5;
}

int a = 1, b = 1, c; // c
doesn't need initializing
TestFunc(a, ref b, out c);
System.Console.WriteLine("
{0} {1} {2}", a, b, c); //
1 2 5

// Accept variable number
of arguments
int Sum(params int[] nums)
{
  int sum = 0;
  foreach (int i in nums)
    sum += i;
  return sum;
}

int total = Sum(4, 3, 2,
1); // returns 10

/* C# doesn't support
optional
arguments/parameters.
```

Functions (cont)

```
Just create two different
versions of the same
function. */
void SayHello(string name,
string prefix) {
  System.Console.WriteLin
e("Greetings, " + prefix
+ " " + name);
}

void SayHello(string name)
{
  SayHello(name, "");
}
```

Structs

```
struct AuthorRecord {
  public string name;
  public float rank;

  public
AuthorRecord(string name,
float rank) {
    this.name = name;
    this.rank = rank;
  }
}
AuthorRecord author = new
AuthorRecord("Steven",
8.8);
AuthorRecord author2 =
author

author.name = "Scott";
SystemConsole.WriteLine(au
thor.name); //Prints
Steven
```

Structs (cont)

```
System.Console.WriteLine(a
uthor2.name); //Prints
Scott
```

Console I/O

```
//Escape sequences
\n, \r
\t
\\
\

Convert.ToChar(65)
//Returns 'A' - equivalent
to Chr(num) in VB
// or
(char) 65

System.Console.Write("What
's your name? ");
string name =
SYstem.Console.ReadLine();
System.Console.Write("How
old are you? ");
int age =
Convert.ToInt32(System.Con
sole.ReadLine());
System.Console.WriteLine("
{0} is {1} years old.",
name, age);
//or
System.Console.WriteLine(n
ame + " is " + age + "
years old.");

int c =
System.Console.Read();
//Read single char
```

Cheatography

See Sharp Cheat Sheet

by Kemmojoo via cheatography.com/28261/cs/8310/

Console I/O (cont)

```
System.Console.WriteLine(c
); //Prints 65 if user
enters "A"
```

Data Types

```
//Value Types
bool
byte, sbyte
char (example: 'A')
short, ushort, int, uint,
long, ulong
float, double
decimal
DateTime
//Reference Types
object
string
int x;
Console.WriteLine(x.GetTyp
e())
Console.WriteLine(typeof(i
nt))

//Type conversion
float d = 3.5;
int i = (int) d
```

Arrays

```
int[] nums = {1, 2, 3};
for (int i = 0; i <
nums.Length; i++)
  Console.WriteLine(nums[i
]);

// 5 is the size of the
array
```

Arrays (cont)

```
string[] names = new
string[5];
names[0] = "Steven";
// Throws
System.IndexOutOfRangeExce
ption
names[5] = "Sarah"

// C# can't dynamically
resize an array.
//Just copy into new
array.
string[] names2 = new
string[7];
// or
names.CopyTo(names2, 0);
Array.Copy(names, names2,
names.Length);

float[,] twoD = new
float[rows, cols];
twoD[2,0] = 4.5;

int[][] jagged = new
int[3][] {
  new int[5], new int[2],
new int[3] };
jagged[0][4] = 5;
```

Classes / Interfaces

```
//Accessibility keywords
public
private
internal
protected
protected internal
static


//Inheritance
```

Classes / Interfaces (cont)

```
class Articles: Authors {
  ...
}

using System;

interface IArticle{
  void Show();
}


class IAuthor:IArticle{
  public void Show() {
    System.Console.WriteLi
ne("Show() method
Implemented");
  }


  public static void
Main(string[] args) {
    IAuthor author = new
IAuthor();
    author.Show();
  }
}
```

File I/O

```
using System.IO;
//Write out to text file
StreamWriter writer =
File.CreateText
  ("c:\\myfile.txt");
writer.WriteLine("Out to
file.");
writer.Close();

//Read all lines from text
file
StreamReader reader =
File.OpenText
```

File I/O (cont)

```
  ("c:\\myfile.txt");
string line =
reader.ReadLine();
while (line != null) {
  Console.WriteLine(line);
  line =
reader.ReadLine();
}
reader.Close();

//Write out to binary file
string str = "Text data";
int num = 123;
BinaryWriter binWriter =
new
BinaryWriter(File.OpenWrit
e
  ("c:\\myfile.dat"));
binWriter.Write(str);
binWriter.Write(num);
binWriter.Close();

//Read from binary file
BinaryReader binReader =
new
BinaryReader(File.OpenRead
  ("c:\\myfile.dat"));
str =
binReader.ReadString();
num =
binReader.ReadInt32();
binReader.Close();
```

Cheatography

See Sharp Cheat Sheet

by Kemmojoo via cheatography.com/28261/cs/8310/

Constants

```
const int MAX_AUTHORS =
25;
readonly float
MIN_RANKING = 5.00;
```

Choices

```
greeting = age < 20 ?
"What's up?" : "Hello";
if (x != 100 && y < 5){
  // Multiple statements
must be enclosed in {}
  x *= 5;
  y *= 2;
}
if (x > 5)
  x *= y;
else if (x == 5)
  x += y;
else if (x < 10)
  x -= y;
else
  x /= y;
//Must be integer or
string
switch (color){
  case "black":
  case "red": r++;
   break;
  case "blue"
   break;
  case "green": g++;
   break;
  default: other++;
   break;
}
```

Exception Handling

```
class Withfinally{
  public static void
Main() {
    try {
      int x = 5;
      int y = 0;
      int z = x/y;
      Console.WriteLine(z)
;
    }
catch(DivideByZeroExceptio
n e) {
      System.Console.Write
Line("Error occurred");
    } finally {
      System.Console.Write
Line("Thank you");
    }
  }
}
```

Constructors / Destructors

```
class TopAuthor {
  private int _topAuthor;

  public TopAuthor() {
     _topAuthor = 0;
  }

  public TopAuthor(int
topAuthor) {
    this._topAuthor=
topAuthor
  }

  ~TopAuthor() {
    // Destructor code to
free unmanaged resources.
    // Implicitly creates
a Finalize method
  }
}
```

Constructors / Destructors (cont)

```
  }
}
```

Properties

```
private int _size;

public int Size {
  get {
    return _size;
  }
  set {
    if (value < 0)
      _size = 0;
    else
      _size = value;
  }
}

foo.Size++;
using System;
class Date{
    public int Day{
        get {
            return day;
        }
        set {
            day = value;
        }
    }
    int day;


    public int Month{
        get {
            return month;
        }
        set {
            month =
value;
```

Properties (cont)

```
        }
    }
    int month;



    public int Year{
        get {
            return year;
        }
        set {
            year = value;
        }
    }
    int year;



    public bool
IsLeapYear(int year) {
        return year%4== 0
? true: false;
    }
    public void SetDate
(int day, int month, int
year) {
        this.day = day;
        this.month =
month;
        this.year = year;
    }
}
```