CS2 Action

// System.Action

/\*

Another new feature introduced by C# 2.0 is the System.Action delegate. An Action is used

by another new C# feature, Array.ForEach( ), to perform an action on each element of an array.

The object to be acted upon is passed in obj. When used with ForEach( ), each element of the

array is passed to obj in turn. Thus, through the use of ForEach( ) and Action, you can, in a

single statement, perform an operation over an entire array.

The following program demonstrates both ForEach( ) and Action. It first creates an

array of Number objects, then uses the method Show( ) to display the values. Next, it uses

Negate( ) to negate the values. Finally, it uses Show( ) again to display the negated values.

These operations

\*/

class Number

{

public int Integer;

public Number(int Integer)

{

this.Integer = Integer;

}

}

class Actions

{

// An Action method.

// Displays the value it is passed.

void Show(Number N)

{

Console.Write(N.Integer + " ");

}

// Another Action method.

// Negates the value it is passed.

void Negate(Number N)

{

N.Integer = -N.Integer;

}

public void Print()

{

Number[] Numbers = new Number[5];

Numbers[0] = new Number(5);

Numbers[1] = new Number(4);

Numbers[2] = new Number(3);

Numbers[3] = new Number(2);

Numbers[4] = new Number(1);

Console.Write("Contents of numbers: ");

// Use action to show the values.

Array.ForEach(Numbers, Show);

Console.WriteLine();

// Use action to negate the values.

Array.ForEach(Numbers, Negate);

Console.Write("Contents of numbers negated: ");

// Use action to negate the values again.

Array.ForEach(Numbers, Show);

Console.WriteLine();

}

}

class Program

{

static void Main()

{

Actions Actn = new Actions();

Actn.Print();

}

}