**Using a variable number of arguments:**

* Sometimes you will want to create a method that can be passed an arbitrary number of arguments.
* For example, consider a method that finds the smallest of a set of values. Such a method might be passed as few as two values, or three, or four, and so on.
* In all cases, you want that method to return the smallest value. Such a method cannot be created using normal parameters. Instead, you must use a special type of parameter that stands for an arbitrary number of parameters.
* This is done by creating a **params** parameter.
* The **params** modifier is used to declare an array parameter that will be able to receive zero or more arguments.
* The number of elements in the array will be equal to the number of arguments passed to the method. Your program then accesses the array to obtain the arguments.
* Here is an example that uses **params** to create a method called **MinVal( )**, which returns the minimum value from a set of values:

// Demonstrate params.

using System;

class Min

{

public int MinVal(params int[] nums)

{

int m;

if(nums.Length == 0)

{

Console.WriteLine("Error: no arguments.");

return 0;

}

m = nums[0];

for(int i=1; i < nums.Length; i++)

if(nums[i] < m) m = nums[i];

return m;

}

}

class ParamsDemo

{

public static void Main()

{

Min ob = new Min();

int min;

int a = 10, b = 20;

// Call with 2 values.

min = ob.MinVal(a, b);

Console.WriteLine("Minimum is " + min);

// Call with 3 values.

min = ob.MinVal(a, b, -1);

Console.WriteLine("Minimum is " + min);

// Call with 5 values.

min = ob.MinVal(18, 23, 3, 14, 25);

Console.WriteLine("Minimum is " + min);

// Can call with an int array, too.

int[] args = { 45, 67, 34, 9, 112, 8 };

min = ob.MinVal(args);

Console.WriteLine("Minimum is " + min);

}

}