**Using ref and out parameters:**

* The use of **ref** and **out** is not limited to the passing of value types. They can also be used when a reference is passed.
* When **ref** or **out** modifies a reference, it causes the reference, itself, to be passed by reference.
* This allows a method to change the object to which the reference refers. Consider the following program, which uses **ref** reference parameters to
* exchange the objects to which two references are referring:

// Swap two references.

using System;

class RefSwap

{

int a, b;

public RefSwap(int i, int j)

{

a = i;

b = j;

}

public void Show()

{

Console.WriteLine("a: {0}, b: {1}", a, b);

}

// This method changes its arguments.

public void Swap(ref RefSwap ob1, ref RefSwap ob2)

{

RefSwap t;

t = ob1;

ob1 = ob2;

ob2 = t;

}

}

class RefSwapDemo

{

public static void Main()

{

RefSwap x = new RefSwap(1, 2);

RefSwap y = new RefSwap(3, 4);

Console.Write("x before call: ");

x.Show();

Console.Write("y before call: ");

y.Show();

Console.WriteLine();

// Exchange the objects to which x and y refer.

x.Swap(ref x, ref y);

Console.Write("x after call: ");

x.Show();

Console.Write("y after call: ");

y.Show();

}

}

**The output from this program is shown here:**

x before call: a: 1, b: 2

y before call: a: 3, b: 4

x after call: a: 3, b: 4

y after call: a: 1, b: 2

* In this example, the method **Swap( )** exchanges the objects to which the two arguments to **Swap( )** refer.
* Before calling **Swap( )**, **x** refers to an object that contains the values 1 and 2, and **y** refers to an object that contains the values 3 and 4.
* After the call to **Swap( )**, **x** refers to the object that contains the values 3 and 4, and **y** refers to the object that contains the values 1 and 2.
* If **ref** parameters had not been used, then the exchange inside **Swap( )** would have had no effect outside **Swap( )**.
* You might want to prove this by removing **ref** from **Swap( )**.