# Review: Array of Simple Types

**The goal for this exercise** is to review how one uses an array of simple types (of value types) in C#. The longer-term goal is to begin preparing to build up a class that manages access to an array for us.

**What you need to do for this exercise:**

1. In the starter project (in the **PCE\_Starter** project), fill in the Array\_Of\_Ints class so that it's RunExercise method will create an array of, say, 10 integers, fill in the array with the first 10 odd values (1, 3, 5, etc), and then print those values out. Make sure that you write your program so that you can change the size of the allocated array, recompile the program, and still have the entire rest of the program fill the array, and print out everything in the array, without modification (mainly, you'll need to the array's Length property, rather than putting in "10" throughout your code).
2. Next, add in a loop to your program. The loop should continue until the user types in "1000". In that loop, ask the user for the space of the array to examine, and print out the contents of that array slot. If the user asks for something that's out of bounds (say, the array is 20 elements long, and the user asks for element 999, or 100, or 20, or -1, or -20, etc), print a message to the user informing them of the error they've made.

The program should not crash, under any circumstances.

You may want to read ahead to the exercise wherein you'll create the SmartArray class, so that you can tailor the work you do here to prepare your better for the SmartArray class.

Quick, sample transcript (user input in **Bold**)

Number to print?  (type in 1000 to quit)

**0**

Number at location 0 is 1

Number to print?  (type in 1000 to quit)

**2**

Number at location 0 is 5

Number to print?  (type in 1000 to quit)

**-10**

I'm sorry, but there is no slot -10

Number to print?  (type in 1000 to quit)

**11**

I'm sorry, but there is no slot 11

 Number to print?  (type in 1000 to quit)

**1000**

Thanks for using the program - have a nice day!