Exercises for Chapter 25

Dynamic

# Exercise 1: Sum an array of arbitrary type

1. Write a method that can sum an arbitrary number of integers. The method should receive a **params int[]** and return an **int** that is the sum of all elements in the array (do not use LINQ operators or extension methods).
2. Convert your method to a generic method that accepts a **params T[]** so that it can be used to calculate the sum of an array of integers, floating-point numbers, or any other type that has the + operator defined.
3. This version of the method will not compile. Why?

(Because there is no constraint to tell the compiler that the **T** generic parameter has a + operator. Unfortunately, there is no such constraint in the CLR implementation of generics.)

1. Now modify the method so that instead of being generic in **T**, it accepts an array of **dynamic**. The method should compile and you should be able to pass to it an array of integers, floating-point numbers, or even strings.
2. Try passing an array of **object** to the method (where the elements of the array are initialized with **new object()**). What happens at compile-time? What happens at runtime?