1. **How do you define a method that takes an arbitrary number of arguments?**

Parameter Array. For example:

public static int min(params int[] paramList)

1. **How do you call a method that takes an arbitrary number of arguments?**

You can call the method by using any number of integer arguments. For example:

int min = Util.Min(first, second)

1. **Why can’t you use an array to pass an arbitrary number of arguments to a method?**

If you use an array to pass an arbitrary number of arguments to a method, the compiler is unable count the correct number of arguments. More specifically, when a method uses the params keyword, the compiler counts the number of arguments in the method call, creates an array of that size, fills in the array with the arguments, and then calls the method by passing the single array parameter.

1. **How many parameters can a method have?**

You can only have one params array per method, and the params array must be the last parameter.

1. **Do parameter arguments have to have the same type?**

No, a method can accept any number of arguments of any type by using the *params* keyword in combination with the *object* type

1. **What is the difference between a method that takes a parameter argument and one that takes optional arguments?**

A method that takes optional arguments is considered overloaded and has a fixed parameter list where you cannot pass an arbitrary list of arguments. Overloaded methods require writing multiple lines of code to account for various numbers and types of arguments.

A method that takes a parameter argument has an arbitrary list of parameters and none of them has a default value. This avoids needing to write additional lines code to account for a large number of overloads since the compiler will count, create, and populate an array for you using the *params* keyword.

**Question 7 on next page…**

1. **How do you define a method that takes different (and arbitrary) types of arguments?**

You can define a method that takes different types of arguments by creating a parameter array of type *object*. For example:

public static void Hole(params object[] paramList)

This technique is based on the fact that the compiler can convert value types to reference types with object boxing.