**Scope and access modifiers are related concepts in programming languages like Java, C++, C#, and others, but they serve different purposes.**

**1. Scope:**  
Scope refers to the visibility or accessibility of a variable, constant, function, or class within a program. It defines where in the code a particular identifier can be accessed. The scope of a variable determines where in the program the variable is valid and can be used.

In most programming languages, there are typically three main types of scope:  
- Global scope: Variables defined in the global scope are accessible from anywhere in the program.  
- Local scope: Variables defined within a function or block are only accessible within that function or block.  
- Class scope: Variables defined within a class are accessible within that class and may have different levels of visibility based on access modifiers.

**2. Access Modifiers:**  
Access modifiers, on the other hand, control the visibility or accessibility of classes, methods, and other members within a class. They define who can access a particular class or member and from where it can be accessed.

Common access modifiers include:  
- Public: Members marked as public are accessible from anywhere, both within and outside the class.  
- Private: Members marked as private are only accessible within the same class.  
- Protected: Members marked as protected are accessible within the same class and its subclasses.  
- Default (package-private in Java): Members with default access are accessible within the same package.

In summary, scope refers to the visibility of variables within a program, while access modifiers control the visibility of classes and members within a class. Scope determines where a variable can be accessed, while access modifiers determine who can access a class or member and from where.