In Java, access modifiers are keywords that are used to control the visibility and accessibility of classes, methods, variables, and other members within a Java program. These modifiers determine which parts of your code can be accessed from other classes or packages and which parts are hidden or restricted.

Java has four main access modifiers:

1. **Public (`public`):** Members declared as public are accessible from anywhere, both within the same class and from other classes and packages. This is the most permissive access modifier.

   ```java
   public class MyClass {
       public int myVariable;
       public void myMethod() {
           // Code here
       }
   }
   ```

2. **Private (`private`):** Members declared as private are only accessible within the class in which they are defined. They cannot be accessed from outside the class, even from subclasses.

   ```java
   public class MyClass {
       private int myVariable;
       private void myMethod() {
           // Code here
       }
   }
   ```

3. **Protected (`protected`):** Members declared as protected are accessible within the same class, subclasses (including subclasses in different packages), and classes in the same package. They are not accessible from classes in different packages that are not subclasses.

   ```java
   public class MyClass {
       protected int myVariable;
       protected void myMethod() {
           // Code here
       }
   }
   ```

4. **Default (Package-private):** If no access modifier is specified (i.e., no `public`, `private`, or `protected` keyword), the member is considered to have package-private access. Members with package-private access are accessible only within the same package.

   ```java
   class MyClass {
       int myVariable;
       void myMethod() {
           // Code here
       }
   }
   ```

It's important to choose the appropriate access modifier for each member based on your design and encapsulation requirements. Using access modifiers helps in achieving encapsulation and ensuring that the internal details of a class are hidden and only the necessary parts are exposed for interaction.