

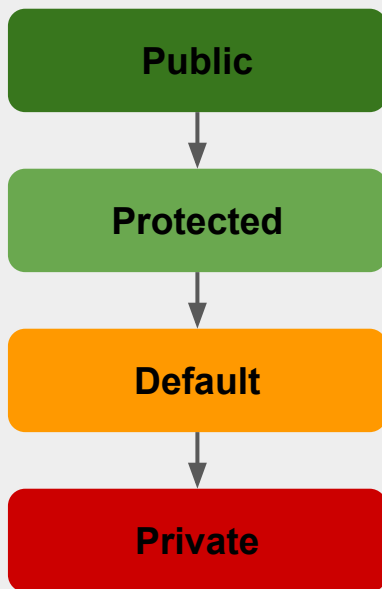
Access Modifiers

Access modifiers

Access modifiers/specifiers in java define the boundary for accessing members of a class and a class itself.

In other words, access modifiers are those modifiers that are used to restrict the visibility (accessibility) of classes, fields, methods, or constructors.

Java provides four explicit access modifiers in object-oriented programming languages.



Public Modifier

- Public access modifier can apply to instance variables, constructors, inner classes, outer class, methods but not with local variables.
- Public members of a class can be used anywhere.
- Public members of a class can be inherited to any subclass.

Protected Modifier

- Protected access modifier can be applied to instance variables, local variables, constructors, methods, inner classes but not the outer class.
- Protected members are accessible inside the class and everywhere within classes in the same package and outside the package but through inheritance only.
- Protected members can be inherited to the subclass.
- If we make constructor as protected then we can create the subclass of that class within the same package but not outside the package

Default Modifier

- When access modifier is not specified to members of a class or a class itself, it is called default access modifier.
- The default can apply to the instance variable, local variable, constructor, methods, inner class, or outer class.
- Default members of a class are visible inside of the class and everywhere within classes in the same package or folder only. Therefore, they can be accessed from outside the classes in the same package but can not be accessed outside the package.
- Default members can be inherited to the subclass within the same package only. It cannot be inherited from outside the package.

Private Modifier

- Private access modifier in java can apply to a variable, method, constructor, inner class but not the outer class that is class itself.
- The instance variable can be private but a local variable cannot be private.
- Private members (field, method, or constructor) of a class cannot be accessed from outside the class. They are accessible only within the class.
- Private members of a superclass cannot be inherited to the subclass. Therefore, they are not accessible in subclasses.
- If we make any constructor as private, we cannot create an object of that class from another class and also cannot create the subclass of that class.
- A class cannot be private except for inner classes. Inner classes are members of the outer class. So, members of the class can be private.
- If we declare a method as private, it behaves as a method declared as final. We cannot call the private method from outside the class.

Access Modifiers in case of Inheritance

- The private members of the superclass cannot be inherited to the subclass because the private members of superclass are not available to the subclass directly. They are only available in their own class.
- The default members of the parent class can be inherited to the derived class within the same package.
- The protected members of a parent class can be inherited to a derived class but the usage of protected members is limited within the package.
- Public members can be inherited to all subclasses.

Interview Questions