

## FAQ's on Access Modifiers

### 1. What are the different types of Access Modifiers?

The following are the types of Access Modifiers:

**Public:** Anything declared as public can be accessed from anywhere.

**Private:** Anything declared as private can't be seen outside of its class.

**Protected:** Anything declared as protected can be accessed by classes in the same package and subclasses in the other packages.

**Default Modifier:** Can be accessed only to classes in the same package.

### 2. State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.

**Public:** Public class is visible in other packages, field is visible everywhere (class must be public too).

**Private:** Private variables or methods may be used only by an instance of the same class that declares the variable or method; A private feature may only be accessed by the class that owns the feature.

**Protected:** Is available to all classes in the same package and also available to all subclasses of the class that owns the protected feature. This access is provided even to subclasses that reside in a different package from the class that owns the protected feature.

### 3. What you get by default, i.e. without any access modifier (i.e. public private or protected).

It means that it is visible to all within a particular package.

### 4. Can a top level class be private or protected?

No. A top level class cannot be private or protected. It can have either "public" or no modifier. If it does not have a modifier it is supposed to have a default access.

If a top level class is declared as private the compiler will complain that the "modifier private is not allowed here". This means that a top level class cannot be private. Same is the case with protected.

### 5. What type of parameter passing does Java support?

In Java the arguments are always passed by value.

### 6. Primitive data types are passed by reference or pass by value?

Primitive data types are passed by value.

### 7. Objects are passed by value or by reference?

Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object.

### 8. Can a class be declared as protected?

A class can't be declared as *protected*. only methods can be declared as *protected*.

### 9. What is the access scope of a protected method?

A *protected* method can be accessed by the classes within the same package or by the subclasses of the class in any package.

### 10. If a variable is declared as private, where may the variable be accessed?

A private variable may only be accessed within the class in which it is declared.

### 11. What do you understand by private, protected and public?

These are accessibility modifiers. *Private* is the most restrictive, while *public* is the least restrictive. There is no real difference between *protected* and the default type (also known as package protected) within the context of the same package; however the protected keyword allows visibility to a derived class in a different package.

### 12. What modifiers may be used with an inner class that is a member of an outer class?

A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

### 13. If a class is declared without any access modifiers, where may the class be accessed?

A class that is declared without any access modifiers is said to have package access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

**14. If a method is declared as protected, where may the method be accessed?**

A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

**15. What is the difference between a public and a non-public class?**

A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

**16. What modifiers may be used with a top-level class?**

A top-level class may be public, abstract, or final.

**17. What modifiers may be used with an inner class that is a member of an outer class?**

A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

**18. What modifiers may be used with a top-level class?**

A top-level class may be public, abstract, or final.