Scala Access Modifier

Access modifier is used to define accessibility of data and our code to the outside world. You can apply accessibly to classes, traits, data members, member methods and constructors etc. Scala provides least accessibility to access to all. You can apply any access modifier to your code according to your application requirement.

Scala provides only three types of access modifiers, which are given below:

No modifier

Protected

Private

In scala, if you don't mention any access modifier, it is treated as no modifier.

Following table contains information about accessbilty of access modifiers.

Modifier	Outside package	Package	Class	Subclass	Companion
No access modifier	Yes	Yes	Yes	Yes	Yes
Protected	No	No	Yes	Yes	Yes
Private	No	No	Yes	No	Yes

Scala Example: Private Access Modifier

In scala, private access modifier is used to make data accessible only within class in which it is declared. It is most restricted and keeps your data in limited scope. Private data members does not inherit into subclasses.

```
class AccessExample{
    private var a:Int = 10
    def show(){
       println(a)
    }
}
object MainObject{
```

```
  def main(args:Array[String]){
    var p = new AccessExample()
    p.a = 12
    p.show()
  }
}
```

Output:

```
error: variable a in class AccessExample cannot be accessed in AccessExample
    p.a = 12
      ^
one error found
```

Scala Example: Protected Access Modifier

Protected access modifier is accessible only within class, sub class and companion object. Data members declared as protected are inherited in subclass. Let's see an example.

```
class AccessExample{
   protected var a:Int = 10
}
class SubClass extends AccessExample{
  def display(){
    println("a = "+a)
  }
}
object MainObject{
  def main(args:Array[String]){
```

```
    var s = new SubClass()
    s.display()
  }
}
```

Output:

a = 10

Scala Example: No-Access-Modifier

In scala, when you don't mention any access modifier, it is treated as no-access-modifier. It is same as public in java. It is least restricted and can easily accessible from anywhere inside or outside the package.

```
class AccessExample{
   var a:Int = 10
   def show(){
      println(" a = "+a)
   }
}
object MainObject{
  def main(args:Array[String]){
    var a = new AccessExample()
    a.show()
  }
}
```

Output:

a = 10