Notes on Access Modifiers in Java

Access Modifiers Overview

1. **Definition**: Control the visibility and accessibility of classes, methods, variables, and constructors in Java.
2. **Purpose**: Protect class members from unauthorized access or modification.
3. **Types of Modifiers**:
   - **Access Modifiers**: private, default, protected, public.
   - **Non-access Modifiers**: final, abstract, synchronized, transient, volatile, strictfp, native.

Access Modifiers

1. **Private**:
   - Applicable to: Instance variables, methods, constructors, inner classes.
   - Not applicable to: Outer classes.
   - Accessible only within the defining class.
   - Cannot be inherited.
   - Private constructors prevent object creation from other classes.
2. **Default**:
   - No explicit keyword used.
   - Applicable to: Instance variables, methods, constructors, classes (outer and inner).
   - Accessible within the same package.
   - Cannot be inherited outside the package.
3. **Protected**:
   - Applicable to: Instance variables, methods, constructors, inner classes.
   - Not applicable to: Outer classes.
   - Accessible within the same package and through inheritance outside the package.
4. **Public**:
   - Applicable to: Instance variables, methods, constructors, inner classes, outer classes.
   - Accessible from anywhere in the program.
   - Can be inherited by any subclass.

Modifier Visibility Summary

Modifier	Class	Package	Subclass (Same Package)	Subclass (Different Package)	World
Private	▯	▯	▯	▯	▯
Default	▯	▯	▯	▯	▯
Protected	▯	▯	▯	▯ (via inheritance)	▯
Public	▯	▯	▯	▯	▯

Modifiers Applicability

1. **Classes**:
   - Outer Classes: public, default, final, abstract, strictfp.
   - Inner Classes: public, private, protected, default, final, abstract, static, strictfp.
2. **Methods**:
   - Applicable: public, protected, private, default, final, static, synchronized, abstract, native, strictfp.
   - Not Applicable: transient, volatile.
3. **Variables**:
   - Applicable: public, protected, private, default, final, static, transient, volatile.
   - Not Applicable: abstract, synchronized, native, strictfp.
4. **Constructors**:
   - Only access modifiers (public, protected, private, default) are applicable.
5. **Blocks**:
   - Modifiers: static, synchronized.
6. **Interfaces**:
   - Outer Interface: public, default, abstract, strictfp.
   - Inner Interface: public, private, protected, default, abstract, static, strictfp.
7. **Enums**:
   - Outer Enum: public, default, strictfp.
   - Inner Enum: public, private, protected, default, static, strictfp.

Key Points

1. Modifier restriction levels: **private > default > protected > public**.
2. Illegal combinations:
   - Final and abstract together on a class.
   - Private or protected alongside public.
3. Non-access modifiers such as `transient`, `volatile`, `synchronized`, etc., are applicable only in specific contexts.

MCQ Question Bank: Access Modifiers in Java

General Concepts

1. **What is the primary purpose of access modifiers in Java?**
   a) To improve code readability
   b) To control visibility and accessibility of classes and members
   c) To execute the code faster

d) To restrict the size of the program
Answer: b

2. **Which of the following is NOT an access modifier in Java?**
   a) private
   b) protected
   c) synchronized
   d) default
   Answer: c

3. **How many access modifiers does Java provide explicitly?**
   a) 2
   b) 3
   c) 4
   d) 5
   Answer: c

Private Modifier

4. **Which of the following statements about private access modifiers is true?**
   a) They can be applied to outer classes.
   b) Private members are accessible within the same package.
   c) Private members are only accessible within the same class.
   d) Private methods can be overridden.
   Answer: c

5. **What happens if a constructor is declared private?**
   a) It can only be called within the same class.
   b) It cannot be called under any circumstance.
   c) It can be inherited by subclasses.
   d) It automatically becomes final.
   Answer: a

Default Modifier

6. **When does a member of a class have default access?**
   a) When no access modifier is specified.
   b) When the `default` keyword is used.
   c) When it is declared static.
   d) When it is declared final.
   Answer: a

7. **Which of the following is true about default access in Java?**
   a) Members are accessible everywhere in the program.
   b) Members are accessible within the same class only.
   c) Members are accessible within the same package.
   d) Members are accessible to subclasses in different packages.
   Answer: c

Protected Modifier

8. **What does the protected modifier allow?**
   a) Access only within the same class.

b) Access from the same package and subclasses in any package.
c) Access from any class in the program.
d) Access only from the superclass.
Answer: b

9. **Which of the following is true about a protected constructor?**
a) It can be called from any package.
b) It can be used to create objects outside the package.
c) It allows creating subclasses in the same package.
d) It cannot be inherited.
Answer: c

Public Modifier

10. **Which of the following statements about public access modifiers is correct?**
a) Public members are only accessible within the same class.
b) Public members cannot be inherited.
c) Public members are accessible from any class in the program.
d) Public access is only allowed for static members.
Answer: c

11. **What is NOT allowed with the public access modifier?**
a) Methods
b) Instance variables
c) Outer classes
d) Local variables
Answer: d

Miscellaneous

12. **Which of the following correctly represents the order of access restriction?**
a) Public > Protected > Default > Private
b) Protected > Private > Default > Public
c) Private > Default > Protected > Public
d) Default > Private > Protected > Public
Answer: c

13. **Which modifiers can be applied to an outer class in Java?**
a) public, private, final
b) public, default, abstract, strictfp
c) public, protected, final, synchronized
d) public, private, strictfp, volatile
Answer: b

14. **Which of the following is NOT applicable to an inner class?**
a) private
b) synchronized
c) strictfp
d) static
Answer: b

15. **What is the only modifier applicable to local variables in Java?**
a) transient

b) final
c) synchronized
d) volatile
Answer: b

This MCQ question bank comprehensively tests the understanding of access modifiers in Java.