Java Modifiers Cheat Sheet

1. Access Modifiers

- public: Accessible from anywhere.

- private: Accessible only within the declared class.

- protected: Accessible in the same package and subclasses.

- default (no modifier): Accessible only in the same package.

2. Non-Access Modifiers

- static: Belongs to the class, not instance.

- final: Cannot be changed, overridden, or inherited (based on context).

- abstract: Must be implemented by subclass (no body).

- synchronized: Used for thread-safe blocks or methods.

- volatile: Prevents thread-local caching of variables.

- transient: Field will not be serialized.

- strictfp: Enforces consistent floating-point precision across platforms.

3. Class-Level Modifiers

- public: Class can be accessed from anywhere.

- final: Class cannot be extended.

- abstract: Class cannot be instantiated.

- strictfp: All FP calculations within follow IEEE standards.

- static: Only allowed for nested classes.

4. Constructor Modifiers

- public: Constructor is accessible everywhere.

- private: Used for singleton or restricted instantiation.

- protected: Useful for inheritance.

- default (no modifier): Package-private access.

5. Illegal Modifier Combinations

- abstract final: Conflicts; can't require subclass and prevent it.

- abstract static (methods): Static methods cannot be abstract.

- private abstract: Abstract must be accessible to subclasses.

6. Summary

Access Modifiers: public, private, protected, default

Non-Access Modifiers: static, final, abstract, synchronized, volatile, transient, strictfp

Use context: classes, methods, fields, constructors, nested classes.