# Access Modifiers Comparison Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modifier** | **C++** | **Python** | **Java** | **Explanation** |
| **Public** | public | No keyword  (public by default) | public | Allows access from any part of the program. In all languages, members declared public are accessible outside the class. |
| **Private** | private | Prefix with \_\_ (double underscore) | private | Restricts access to the class itself. Only the class members can access private members. |
| **Protected** | protected | Prefix with \_ (single underscore) | protected | Allows access within the class and its derived classes (subclasses). |
| **Default** | No specific keyword  (Package-private) | No keyword | No keyword (Packageprivate) | Accessible within the same package (Java). In C++, if no access modifier is provided, members are private by default. |

# Java Access Modifier Comparison (Classes vs. Variables)

|  |  |  |  |
| --- | --- | --- | --- |
| **Access Modifier** | **Class** | **Variable** | **Explanation** |
| **Public** | Class is accessible from anywhere | Variable is accessible from any class | Both the class and variables marked as public are accessible from any other class or package. |
| **Private** | Not allowed for top-level classes | Variable is accessible only within the class | Top-level classes cannot be private. For variables, private restricts access to within the defining class only. |
| **Protected** | Not allowed for top-level classes | Variable is accessible within package or subclass | protected is valid for variables but not for top-level classes. Variables marked as protected are accessible in subclasses and within the package. |
| **Default (Package private)** | Class is accessible within the package only | Variable is accessible within the package | If no access modifier is specified, both classes and variables are accessible only within the same package (Package-private). |

# C++ Access Modifier Comparison (Classes vs. Variables)

|  |  |  |  |
| --- | --- | --- | --- |
| **Access**  **Modifier** | **Class** | **Variable** | **Explanation** |
| **Public** | Class is accessible from anywhere | Variable is accessible from anywhere | Both classes and variables marked as public are accessible from any part of the code. |
| **Private** | Not allowed for top-level classes | Variable is accessible only within the class | Top-level classes cannot be private. Variables marked as private are accessible only within the class. |
| **Protected** | Not allowed for top-level classes | Variable is accessible within the class and subclasses | protected is valid for variables but not for toplevel classes. Variables marked as protected are accessible within the class and its subclasses. |
| **Default** | Class is private if no modifier is provided | Variable is private if no modifier is provided | By default, both classes and variables are private if no access modifier is specified. |

# Python Access Modifier Comparison (Classes vs. Variables)

|  |  |  |  |
| --- | --- | --- | --- |
| **Access**  **Modifier** | **Class** | **Variable** | **Explanation** |
| **Public** | All classes are public by default | Variable is accessible from anywhere | In Python, both classes and variables are public by default, meaning they can be accessed from any part of the program. |
| **Private** | Not applicable  (no strict private) | Variable prefixed with \_\_ is private | Classes can't be private in Python, but variables prefixed with \_\_ are considered private and name-mangled to avoid external access. |
| **Protected** | Convention: prefix with \_ | Variable prefixed with \_ is considered protected | Both classes and variables can follow the \_ convention to indicate that they should be treated as protected (not for external use). |
| **Default** | Public by default | Public by default | All classes and variables are public by default unless they follow the \_\_ (private) or \_ (protected) conventions. |

**Key Takeaways:**

* **Java** has the most granular control over access modifiers, allowing different levels of access for both classes and variables.
* **C++** has similar rules for variables as Java, but top-level classes cannot be marked as private or protected.
* **Python** uses conventions (\_ for protected and \_\_ for private) rather than strict access control mechanisms, making its approach more flexible but less strict