Access and Non Access Modifiers in Java

Difficulty Level : Easy Last Updated : 27 Jan, 2022

Java provides a rich set of modifiers. They are used to control access mechanisms and also provide information about class functionalities to JVM. They are divided into two categories namely:

1. Access modifiers
2. Non-access modifiers

Type 1: Access Modifiers

Java's access modifiers are **public**, **private**, and **protected**. Java also defines a default access level (called package-private). Let us do discuss how they work prior to which go through a deeper level of understanding of them.

- **public:** When a member of a class is modified by **public**, then that member can be accessed by any other code.
- **private:** When a member of a class is specified as **private**, then that member can only be accessed by other members of its class.
  Now you can understand why main( ) has always been preceded by the public modifier. It is called by code that is outside the program—that is, by the Java run-time system. When no access modifier is used, then by default the member of a class is public within its own package, but cannot be accessed outside of its package. **protected** applies only when inheritance is involved.

Type: 2: Non-access Modifiers

- In java, we have 7 non-access modifiers. They are used with classes, methods, variables, constructors etc to provide information about their behavior to JVM. They are as follows:
  1. static
  2. final
  3. abstract
  4. synchronized
  5. transient
  6. volatile
  7. native

This article is contributed by **Gaurav Miglani**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.