Java Keywords and its functions

1. abstract – Specifies that a class or method is abstract (not fully implemented).
2. assert – Used for debugging to test assumptions.
3. boolean – Declares a variable with true/false values.
4. break – Exits from a loop or switch statement.
5. byte – Defines an 8-bit integer variable.
6. case – Defines a branch in a switch statement.
7. catch – Handles exceptions thrown by try blocks.
8. char – Defines a single 16-bit Unicode character variable.
9. class – Declares a class.
10. const – Reserved (not used).
11. continue – Skips the current iteration in a loop.
12. default – Specifies the default branch in a switch statement.
13. do – Executes a loop body at least once (do-while).
14. double – Defines a double-precision floating-point variable.
15. else – Specifies the alternative block in an if statement.
16. enum – Declares an enumeration (fixed set of constants).
17. extends – Indicates inheritance between classes.
18. final – Marks a variable, method, or class as unchangeable.
19. finally – Defines a block that always executes after try-catch.
20. float – Defines a single-precision floating-point variable.
21. for – Defines a loop that runs a fixed number of times.
22. goto – Reserved (not used).
23. if – Tests a condition and executes code if true.
24. implements – Specifies that a class implements an interface.
25. import – Brings other packages or classes into scope.
26. instanceof – Tests whether an object is an instance of a class.
27. int – Defines a 32-bit integer variable.
28. interface – Declares an interface.
29. long – Defines a 64-bit integer variable.
30. native – Specifies that a method is implemented in native code (like C/C++).
31. new – Creates new objects.
32. null – Represents the absence of any object reference.

33. package – Defines a namespace for classes.
34. private – Access modifier: visible only within the same class.
35. protected – Access modifier: visible within package and subclasses.
36. public – Access modifier: visible everywhere.
37. return – Exits from a method and optionally returns a value.
38. short – Defines a 16-bit integer variable.
39. static – Declares members that belong to the class, not instances.
40. strictfp – Enforces strict floating-point calculations.
41. super – Refers to the superclass or calls its constructor.
42. switch – Executes code based on matching a value.
43. synchronized – Prevents concurrent access by multiple threads.
44. this – Refers to the current object.
45. throw – Throws an exception.
46. throws – Declares exceptions that a method might throw.
47. transient – Prevents serialization of a field.
48. try – Starts a block of code to test for exceptions.
49. void – Specifies that a method does not return a value.
50. volatile – Marks a variable as being accessed by multiple threads.
51. while – Defines a loop that continues while a condition is true.
52. var – (Java 10+) Local variable type inference.
53. record – (Java 14+) Declares a record (data-carrying class).
54. sealed – (Java 17+) Restricts which classes can extend a class.
55. permits – (Java 17+) Lists allowed subclasses of a sealed class.
56. _ – (Java 9+) Reserved keyword (cannot be used as an identifier).
57. exports – (Java 9+) Used in modules to export packages.
58. module – (Java 9+) Defines a module.
59. open – (Java 9+) Allows deep reflection on packages.
60. opens – (Java 9+) Opens packages for reflective access.
61. provides – (Java 9+) Declares a service provider in a module.
62. requires – (Java 9+) Specifies module dependencies.
63. to – (Java 9+) Used in module directives (exports ... to).
64. uses – (Java 9+) Declares a service that a module uses.
65. with – (Java 9+) Used in module declarations (provides ... with).
66. yield – (Java 13+) Returns a value from a switch expression.
67. true, false – Boolean literals (not technically keywords, but reserved).