1. **abstract** – Used for a class or method that is not complete.
2. **assert** – Used to test if something is true.
3. **boolean** – Holds true or false values.
4. **break** – Stops a loop or switch.
5. **byte** – Stores small numbers (8 bits).
6. **case** – A label inside a switch.
7. **catch** – Handles errors in a try block.
8. **char** – Holds one character.
9. **class** – Used to define a class.
10. **const** – Reserved word, not used.
11. **continue** – Skips to the next loop step.
12. **default** – Runs when no other case matches in switch.
13. **do** – Runs a loop once before checking the condition.
14. **double** – Holds decimal numbers (64-bit).
15. **else** – Runs when “if” is false.
16. **enum** – Used to make a list of constant values.
17. **extends** – A class inherits from another class.
18. **final** – Used for constants or to stop change or inheritance.
19. **finally** – Always runs after try/catch.
20. **float** – Holds small decimal numbers (32-bit).
21. **for** – Used to make a loop.
22. **goto** – Reserved word, not used.
23. **if** – Tests a condition.
24. **implements** – A class uses an interface.
25. **import** – Brings in other packages or classes.
26. **instanceof** – Checks if an object belongs to a class.
27. **int** – Holds whole numbers (32-bit).
28. **interface** – Used to define abstract methods.
29. **long** – Holds big whole numbers (64-bit).
30. **native** – Calls code written in another language.
31. **new** – Creates a new object.
32. **package** – Groups related classes.
33. **private** – Can be used only inside the same class.
34. **protected** – Can be used in the same package or subclass.
35. **public** – Can be used anywhere.
36. **return** – Sends a value back from a method.
37. **short** – Holds small whole numbers (16-bit).
38. **static** – Belongs to the class, not an object.
39. **strictfp** – Makes floating numbers the same on all computers.
40. **super** – Refers to the parent class.
41. **switch** – Chooses one block of code to run.
42. **synchronized** – Controls access by many threads.
43. **this** – Refers to the current object.
44. **throw** – Sends an exception.
45. **throws** – Declares possible exceptions.
46. **transient** – Skips a field when saving an object.
47. **try** – Tests a block of code for errors.
48. **void** – Means the method returns nothing.
49. **volatile** – Used in multi-threading for shared variables.
50. **while** – Runs a loop while a condition is true.
51. **open** – Opens a package for other modules.
52. **requires** – Adds another module as a need.
53. **exports** – Shares a package with other modules.
54. **opens** – Lets other modules see package contents.
55. **provides** – Declares a service provider.
56. **uses** – Declares a service user.
57. **transitive** – Re-exports a module dependency.
58. **var** – Lets Java guess the type of a local variable.
59. **yield** – Returns a value from a switch expression.
60. **record** – Creates a simple data class.
61. **sealed** – Limits which classes can extend it.
62. **non-sealed** – Allows a sealed class to be extended.
63. **permits** – Lists which classes can extend a sealed class.