

1. Global - accessible from anywhere  
 2. Static - cannot be changed  
 3. Final - cannot be overridden  
 4. Abstract - cannot be instantiated  
 5. Interface - a class that only defines methods  
 6. Annotation - a marker for the compiler  
 7. Enum - a class that represents a set of values  
 8. Var - a variable that can hold any type of data  
 9. String - a sequence of characters  
 10. Integer - a whole number  
 11. Double - a floating point number  
 12. Boolean - a true or false value  
 13. Character - a single character  
 14. Byte - a small integer  
 15. Short - a small integer  
 16. Long - a large integer  
 17. Float - a floating point number  
 18. BigDecimal - a decimal number  
 19. BigInteger - a large integer  
 20. Math - a class that contains mathematical constants and functions  
 21. System - a class that contains system properties and methods  
 22. Object - the root of the class hierarchy  
 23. Class - a blueprint for creating objects  
 24. Object - a collection of objects  
 25. Array - a collection of elements of the same type  
 26. Vector - a collection of objects that can grow in size  
 27. LinkedList - a collection of objects that are linked together  
 28. HashSet - a collection of unique objects  
 29. HashMap - a collection of objects that are mapped to keys  
 30. TreeMap - a collection of objects that are sorted by key  
 31. PriorityQueue - a collection of objects that are sorted by priority  
 32. Stack - a collection of objects that are added and removed in a last-in, first-out manner  
 33. Queue - a collection of objects that are added and removed in a first-in, first-out manner  
 34. Deque - a collection of objects that can be added and removed from both ends  
 35. Iterator - an object that allows you to iterate over a collection of objects  
 36. Iterable - an interface that defines the methods for iterating over a collection of objects  
 37. Comparable - an interface that defines the methods for comparing objects  
 38. Comparator - an interface that defines the methods for comparing objects  
 39. Runnable - an interface that defines the methods for running a task  
 40. Callable - an interface that defines the methods for calling a task  
 41. Future - an interface that defines the methods for getting the result of a task  
 42. Executor - an interface that defines the methods for executing a task  
 43. ExecutorService - an interface that defines the methods for executing a task  
 44. ThreadPoolExecutor - a class that implements the ExecutorService interface  
 45. Executors - a class that provides methods for creating ExecutorService objects  
 46. FutureTask - a class that implements the Future interface  
 47. CountDownLatch - a class that allows you to wait for a set of tasks to complete  
 48. CyclicBarrier - a class that allows you to wait for a set of tasks to complete  
 49. Semaphore - a class that allows you to limit the number of threads that can access a resource  
 50. ReentrantLock - a class that allows you to lock a resource