1. **Which of the following is NOT a part of the Java Collections Framework?**
   a) `ArrayList`
   b) `HashSet`
   c) `TreeMap`
   d) `Array`
2. **Which List implementation is best for frequent insertions and deletions in the middle?**
   a) `ArrayList`
   b) `LinkedList`
   c) `Vector`
   d) `Stack`
3. **What is the primary characteristic of a `Set` in Java?**
   a) Allows duplicate elements
   b) Maintains insertion order
   c) Does not allow duplicate elements
   d) Sorts elements automatically
4. **Which Map implementation maintains keys in sorted order?**
   a) `HashMap`
   b) `LinkedHashMap`
   c) `TreeMap`
   d) `Hashtable`
5. **What is the default initial capacity of an `ArrayList` in Java?**
   a) 5
   b) 10
   c) 16
   d) 20

6. **What is the main advantage of using Generics in Java?**
   a) Improves performance
   b) Ensures type safety at compile time

c) Reduces memory usage

d) Allows dynamic method dispatch

7. **Which of the following is a correct way to declare a generic class?**

   a) `class Box<T> { ... }`

   b) `class Box<T extends Object> { ... }`

   c) `class Box<generic T> { ... }`

   d) `class Box<T super Object> { ... }`

8. **What happens if you try to add a `String` to a `List<Integer>`?**

   a) Compiles and runs successfully

   b) Compile-time error

   c) Runtime exception

   d) The `String` is automatically converted to `Integer`

9. **What is type erasure in Java Generics?**

   a) Removing generic type information at runtime

   b) Converting generic types to specific types

   c) Preventing the use of raw types

   d) Enforcing type constraints at runtime

10. **Which wildcard (`?`) represents an unknown type that is a subclass of a specific class?**

    a) `<?>`

    b) `<? extends T>`

    c) `<? super T>`

    d) `<? implements T>`

Working with Lambda Expressions and Functional Interfaces for Concise Code

11. **What is a functional interface in Java?**

    a) An interface with multiple abstract methods

    b) An interface with a single abstract method

c) An interface with default methods only

d) An interface marked with `@Functional`

12. **Which of the following is a valid lambda expression in Java?**

    a) `(int x) -> { return x * x; }`

    b) `x -> return x * x`

    c) `(x) -> return x * x`

    d) `int x -> x * x`

13. **What does the `Predicate<T>` functional interface represent?**

    a) A function that takes no arguments and returns `T`

    b) A function that takes `T` and returns a boolean

    c) A function that takes two arguments and returns `T`

    d) A consumer of `T`

14. **Which method reference is equivalent to `(String s) -> s.length()`?**

    a) `String::length`

    b) `s::length`

    c) `String.length()`

    d) `length(String)`

15. **What is the purpose of the `forEach` method in Java streams?**

    a) To filter elements

    b) To perform an action on each element

    c) To reduce elements to a single value

    d) To sort elements

16. **Which class is used to represent a date without time in Java 8+?**

    a) `java.util.Date`

    b) `java.time.LocalDate`

    c) `java.sql.Date`

    d) `java.time.Instant`

17. **What does `LocalDateTime` represent?**

    a) A date only

b) A time only

c) A date and time without a timezone

d) A timestamp with timezone

18. **Which method is used to add days to a** `LocalDate`**?**

    a) `addDays()`

    b) `plusDays()`

    c) `appendDays()`

    d) `withDay()`

19. **What is the output of** `Period.between(LocalDate.of(2023, 1, 1), LocalDate.of(2023, 2, 1)).getMonths()`**?**

    a) 0

    b) 1 *(Correct Answer)*

    c) 30

    d) 31

20. **Which class is used to format dates in Java 8+?**

    a) `SimpleDateFormat`

    b) `DateTimeFormatter`

    c) `DateFormat`

    d) `DateFormatter`

21. **Which JDBC method is used to execute a SQL** `SELECT` **query?**

    a) `executeUpdate()`

    b) `executeQuery()`

    c) `execute()`

    d) `executeSelect()`

22. **What is the purpose of** `PreparedStatement` **in JDBC?**

    a) To execute stored procedures

    b) To prevent SQL injection by precompiling SQL

    c) To batch multiple SQL statements

    d) To create database connections

23. **Which JDBC interface represents a connection to the database?**
    a) `Statement`
    b) `ResultSet`
    c) `Connection`
    d) `DriverManager`
24. **What is the correct order of steps in a JDBC workflow?**
    a) Load Driver → Get Connection → Execute Query → Close Connection
    b) Get Connection → Load Driver → Execute Query → Close Connection
    c) Execute Query → Load Driver → Get Connection → Close Connection
    d) Close Connection → Load Driver → Get Connection → Execute Query
25. **Which method is used to retrieve a `ResultSet` after executing a query?**
    a) `Statement.getResult()`
    b) `Statement.executeResult()`
    c) `Statement.getResultSet()`
    d) `Statement.fetchResult()`

26. **What is the purpose of Mockito in unit testing?**
    a) To create real database connections
    b) To mock dependencies and test behavior
    c) To automate GUI testing
    d) To measure code coverage
27. **Which annotation is used to create a mock object in Mockito?**
    a) `@Mock`
    b) `@InjectMocks`
    c) `@Spy`
    d) `@Test`
28. **What does `when(mock.method()).thenReturn(value)` do?**
    a) Calls the real method
    b) Defines mock behavior for a method call

c) Throws an exception

d) Verifies method invocation

29. **Which Mockito method is used to verify that a method was called?**

    a) `verify()`

    b) `assertCalled()`

    c) `checkInvocation()`

    d) `confirm()`

30. **What is the purpose of `@InjectMocks`?**

    a) To create a mock object

    b) To inject mocked dependencies into the tested class

    c) To ignore a test method

    d) To reset all mocks

Error Handling and Debugging Techniques for Efficient Problem-Solving

31. **Which keyword is used to explicitly throw an exception in Java?**

    a) `throws`

    b) `throw`

    c) `catch`

    d) `finally`

32. **What is the purpose of the `finally` block in exception handling?**

    a) To handle exceptions

    b) To execute code regardless of exceptions

    c) To declare exceptions

    d) To skip exception handling

33. **Which exception is thrown when dividing an integer by zero?**

    a) `NullPointerException`

    b) `ArithmeticException`

    c) `IllegalArgumentException`

    d) `NumberFormatException`

34. **What is the output of** `System.out.println(10 / 0);`**?**
    a) `Infinity`
    b) `0`
    c) Compilation error
    d) `ArithmeticException` at runtime
35. **Which debugging technique allows stepping through code line by line?**
    a) Logging
    b) Breakpoints in an IDE
    c) Print statements
    d) Exception handling