

# Advanced Java (JDBC, Servlets, JSP, Spring Boot, Angular) - 100 MCQ Test with Explanations

Time: 120 minutes | Questions: 100 | Marking: +1 / 0 | Includes detailed explanations after each answer.

1. Which interface is used to execute parameterized SQL queries?

- A. Statement
- B. PreparedStatement
- C. CallableStatement
- D. ResultSet

Answer: **B**

PreparedStatement precompiles SQL with placeholders (?) and binds values safely, improving performance and preventing SQL injection.

2. What does executeQuery() return?

- A. boolean
- B. int
- C. ResultSet
- D. void

Answer: **C**

executeQuery() is used for SELECT statements and returns a ResultSet cursor over the rows.

3. Which method disables auto-commit in JDBC?

- A. setCommit(false)
- B. disableCommit()
- C. setAutoCommit(false)
- D. commit(false)

Answer: **C**

Transactions begin when auto-commit is turned off via Connection.setAutoCommit(false).

4. Batch processing primarily uses which methods?

- A. addBatch() and executeBatch()
- B. commit() and rollback()
- C. close() and flush()
- D. prepare() and run()

Answer: **A**

You add multiple parameter sets with addBatch() and send them to the DB in one go using executeBatch().

5. Which JDBC driver type is pure Java and platform independent?

- A. Type 1

- B. Type 2
- C. Type 3
- D. Type 4

Answer: **D**

Type 4 is a thin driver that speaks the DB's wire protocol directly using pure Java, so no native libraries are needed.

6. Which interface is used to call stored procedures?

- A. PreparedStatement
- B. CallableStatement
- C. Statement
- D. Driver

Answer: **B**

CallableStatement supports calling procedures/functions using {call ...} syntax and registering OUT parameters.

7. Which object represents a database session/connection?

- A. DriverManager
- B. Connection
- C. SessionFactory
- D. ResultSet

Answer: **B**

Connection encapsulates a live link to the DB, transactional state, and isolation level.

8. How do you prevent SQL Injection effectively in JDBC?

- A. Use Statement
- B. Use PreparedStatement with bind parameters
- C. Use string concatenation
- D. Turn off autoCommit

Answer: **B**

Bind parameters are sent separately from SQL text, so user input can't alter the query structure.

9. What does ResultSet.next() return when there is no next row?

- A. -1
- B. null
- C. false
- D. true

Answer: **C**

next() advances the cursor and returns false when the cursor moves past the last row.

10. Which cursor type allows both forward and backward movement?

- A. TYPE\_FORWARD\_ONLY
- B. TYPE\_SCROLL\_INSENSITIVE
- C. TYPE\_HOLD\_CURSORS\_OVER\_COMMIT
- D. CONCUR\_READ\_ONLY

Answer: **B**

TYPE\_SCROLL\_INSENSITIVE allows scrolling without reflecting concurrent DB changes.

11. Which concurrency mode allows updates to the ResultSet?

- A. CONCUR\_READ\_ONLY
- B. CONCUR\_UPDATABLE
- C. TYPE\_SCROLL\_SENSITIVE
- D. TYPE\_FORWARD\_ONLY

Answer: **B**

CONCUR\_UPDATABLE enables updating rows directly through the ResultSet APIs.

12. Which exception is commonly thrown by JDBC operations?

- A. IOException
- B. SQLException
- C. RuntimeException
- D. TimeoutException

Answer: **B**

JDBC maps DB and driver errors to java.sql.SQLException and subclasses.

13. How do you set transaction boundaries in JDBC?

- A. Use Connection.setAutoCommit(false), commit(), rollback()
- B. Use Statement.begin() and end()
- C. Use DriverManager.start()
- D. Use ResultSet.commit()

Answer: **A**

Transactions are controlled at Connection level: disable auto-commit, then explicitly commit or rollback.

14. What is connection pooling primarily used for?

- A. Reducing SQL size
- B. Reusing database connections to improve performance
- C. Encrypting connections
- D. Caching ResultSets

Answer: **B**

Creating TCP connections is expensive; pools keep a ready set of connections to reduce latency.

15. Which is the fastest widely-used JDBC connection pool?

- A. C3P0
- B. HikariCP
- C. BoneCP
- D. DBCP v1

Answer: **B**

HikariCP is optimized for low-latency and minimal overhead in high-throughput apps.

16. Which method retrieves generated keys after an INSERT?

- A. getGeneratedKeys()
- B. getAutoKeys()
- C. getKeys()
- D. fetchKeys()

Answer: **A**

Use Statement/PreparedStatement with RETURN\_GENERATED\_KEYS and then call getGeneratedKeys().

17. What is the correct placeholder in PreparedStatement?

- A. \$1
- B. :id
- C. ?
- D. #{}  
}

Answer: **C**

JDBC uses positional placeholders (?) and sets them 1-based via setXxx(index, value).

18. How do you register a JDBC driver in modern Java?

- A. Class.forName() is always mandatory
- B. Driver auto-loading via SPI
- C. Edit the JRE lib
- D. It is not needed to load drivers at all

Answer: **B**

Since JDBC 4.0, drivers auto-register via META-INF/services (Service Provider Interface).

19. Which method on PreparedStatement executes DML and returns affected

- rows?
- A. executeQuery()
  - B. executeUpdate()
  - C. execute()
  - D. run()

Answer: **B**

`executeUpdate()` returns the update count for INSERT/UPDATE/DELETE (or 0 for statements like DDL).

20. What does `setFetchSize(int)` influence?

- A. Maximum rows in table
- B. Batch size of network fetch from DB
- C. Number of connections in pool
- D. Transaction isolation

Answer: **B**

Drivers may use fetch size as a hint for how many rows to pull per roundtrip for a `ResultSet`.

21. Which class do most HTTP Servlets extend?

- A. GenericServlet
- B. HttpServlet
- C. Servlet
- D. Filter

Answer: **B**

`HttpServlet` provides `doGet/doPost` and HTTP-specific conveniences on top of `GenericServlet`.

22. Which method is invoked once during the servlet lifecycle?

- A. `service()`
- B. `init()`
- C. `doGet()`
- D. `doPost()`

Answer: **B**

`init()` is called once to initialize the servlet; `service()` runs per request.

23. Which object is used to read request parameters?

- A. `HttpServletRequest`
- B. `ServletConfig`
- C. `HttpServletRequest`
- D. `ServletContext`

Answer: **C**

`HttpServletRequest` exposes `getParameter()`, `getHeader()`, `getSession()`, etc.

24. Which method writes character data in response?

- A. `getOutputStream()`
- B. `getWriter()`
- C. `writeText()`
- D. `getResponseText()`

Answer: **B**

`getWriter()` returns a `PrintWriter` for text; `getOutputStream()` is for binary output.  
25. `sendRedirect()` performs a:

- A. Server-side forward
- B. Client-side redirect
- C. Filter mapping
- D. Error dispatch

Answer: **B**

It sends a 3xx status + Location header; the browser makes a new request.

26. `RequestDispatcher.forward()` results in:

- A. New request from client
- B. Same request forwarded on server
- C. Browser URL changes
- D. Session invalidation

Answer: **B**

Forward happens on the server with the same request/response objects and URL stays the same.

27. Which scope is shared across the entire application?

- A. request
- B. session
- C. application
- D. page

Answer: **C**

`ServletContext` (application scope) is shared across all sessions and requests.

28. Which header best prevents caching sensitive responses?

- A. Content-Type
- B. Cache-Control: no-store
- C. Accept
- D. ETag

Answer: **B**

'no-store' instructs caches not to store any part of the response.

29. What is a Filter used for?

- A. Compiling JSP
- B. Pre/post-processing requests and responses
- C. Handling WebSocket only

- D. Creating cookies only

Answer: **B**

Filters wrap around resources to add cross-cutting concerns (auth, logging, compression).

30. Which method to retrieve an HTTP session (create if absent)?

- A. request.getSession()
- B. request.getContext()
- C. response.getSession()
- D. context.getSession()

Answer: **A**

getSession() with no args creates one if not present; getSession(false) avoids creation.

31. URL rewriting appends session id using which parameter by default?

- A. JSESSIONID
- B. SESSION
- C. SID
- D. SSID

Answer: **A**

Most servlet containers use the JSESSIONID parameter when cookies are disabled.

32. Which status code is typical for redirect?

- A. 200
- B. 301/302
- C. 404
- D. 500

Answer: **B**

Temporary or permanent redirects use 302/301 (and other 3xx codes).

33. Which annotation maps a servlet without web.xml?

- A. @WebFilter
- B. @WebInitParam
- C. @WebServlet
- D. @MultipartConfig

Answer: **C**

@WebServlet declares URL patterns for a servlet class in annotation style.

34. Multipart file upload is enabled using:

- A. @EnableUpload
- B. @MultipartConfig

C. @UploadServlet

D. @WebMultipart

Answer: **B**

Annotating the servlet with `@MultipartConfig` lets it parse multipart/form-data requests.

35. Which is BEST to store small user-level data across requests?

A. ServletContext attributes

B. HttpSession attributes

C. Request attributes

D. Hidden fields only

Answer: **B**

`HttpSession` persists per user across multiple requests; request attributes die after one request.

36. Which method returns query string parameter value?

A. `request.getParameter(name)`

B. `request.getAttribute(name)`

C. `session.getAttribute(name)`

D. `context.getInitParameter(name)`

Answer: **A**

`getParameter()` reads from the query string (GET) or request body (POST with form encoding).

37. To forward to JSP from servlet you use:

A. `response.sendRedirect()`

B.

`request.getRequestDispatcher('/view.jsp').forward(request,response)`

`@WebForward`

D. `JspFactory.forward()`

Answer: **B**

Forward avoids a round-trip and keeps request attributes available for the JSP.

38. Thread safety issue in servlets commonly caused by:

A. Local variables

B. Instance fields shared across requests

C. Final constants

D. Static methods only

Answer: **B**

Container creates one servlet instance; mutable instance fields are accessed by many threads.

39. Default HTTP method handled if not overridden:

- A. doService()
- B. service()
- C. doGet()
- D. handle()

Answer: **B**

service() dispatches to doGet/doPost etc.; if not overridden, HttpServlet may return 405.  
40. Which interface gives access to app-wide init params?

- A. ServletConfig
- B. ServletContext
- C. FilterConfig
- D. HttpSessionContext

Answer: **B**

ServletContext stores application-level initialization parameters and attributes.

41. JSPs are compiled into:

- A. JavaScript
- B. Servlet classes
- C. WAR files
- D. HTML only

Answer: **B**

JSPs are translated into servlets (Java classes) and then compiled by the container.

42. Which is NOT an implicit JSP object?

- A. request
- B. response
- C. application
- D. database

Answer: **D**

'database' is not implicit; JDBC access is done via code/DAOs.

43. Which scripting element outputs expression value?

- A. <%! %>
- B. <% %>
- C. <%= %>
- D. <%@ %>

Answer: **C**

<%= %> evaluates and prints the expression to the JspWriter.

44. Which directive includes another file at translation time?

- A.
- B. <%@ include file=... %>
- C.
- D.

Answer: **B**

The include directive is static (happens before compilation); is dynamic at request time.

45. Which implicit object refers to ServletContext?

- A. pageContext
- B. config
- C. application
- D. session

Answer: **C**

'application' is a shortcut to the ServletContext from a JSP.

46. To handle exceptions in JSP use:

- A. errorPage directive and isErrorPage
- B. try/catch only
- C.
- D. ServletFilter

Answer: **A**

Use <%@ page errorPage='err.jsp' %> and on err.jsp declare isErrorPage='true'.

47. Expression Language (EL) syntax for attribute 'name' in request scope is:

- A. \${request.name}
- B. \${requestScope.name}
- C. \${name}
- D. All of the above

Answer: **D**

EL resolves variables by scope; all shown notations can reference request-scoped attribute 'name'.

48. JSTL core tag library prefix is commonly:

- A. c
- B. jstl
- C. core
- D. t

Answer: **A**

The core library is typically declared as prefix 'c' (e.g., `c.` ).

49. Which implicit object gives JspWriter?

- A. out
- B. writer
- C. printer
- D. response

Answer: **A**

'out' is the JspWriter used for buffered output in JSP.

50. Setting content type in JSP is done via:

- A. `<%@ page contentType=... %>`
- B.
- C. `<% setType() %>`
- D. `response.setContentType()`

Answer: **A**

The page directive controls page attributes like `contentType` and `pageEncoding`.

51. Which scope persists until user session invalidation?

- A. page
- B. request
- C. session
- D. application

Answer: **C**

Session scope lasts across requests for a user until timeout or invalidation.

52. Which tag forwards control at request time?

- A. `<%@ include %>`
- B.
- C.
- D.

Answer: **C**

transfers control to another resource during the same request.

53. EL disables scriptlets and encourages:

- A. static HTML only
- B. MVC separation and cleaner templates
- C. binary JSPs
- D. deprecated features

Answer: **B**

EL + JSTL keeps Java code out of JSP, supporting clean MVC views.

54. The page directive is written as:

- A. <%@ page ... %>
- B.
- C. <% page %>
- D.

Answer: **A**

Directives use <%@ ... %> syntax; 'page' sets page-level options.

55. To use a bean in JSP with id 'u' use:

- A.
- B. u
- C. <%@ bean u %>
- D. <% bean u %>

Answer: **A**

instantiates or locates an existing bean in a given scope.

56. @SpringBootApplication combines:

- A. @EnableAutoConfiguration, @ComponentScan, @Configuration
- B. @Controller, @Service, @Repository
- C. @Entity, @Table, @Id
- D. @RestController, @Bean, @Profile

Answer: **A**

This meta-annotation enables auto-config, component scanning in the package, and registers configuration classes.

57. Which starter for REST web apps?

- A. spring-boot-starter-web
- B. spring-boot-starter-aop
- C. spring-boot-starter-mail
- D. spring-boot-starter-actuator

Answer: **A**

The web starter brings Spring MVC, Jackson, and an embedded server for REST endpoints.

58. Default embedded server is:

- A. Jetty
- B. Undertow
- C. Tomcat

D. Netty

Answer: **C**

Spring Boot defaults to Tomcat but supports switching to Jetty/Undertow via dependencies.

59. Which annotation defines a REST controller?

- A. `@Controller`
- B. `@RestController`
- C. `@ResponseBody`
- D. `@ControllerAdvice`

Answer: **B**

`@RestController` = `@Controller` + `@ResponseBody`, returning JSON/XML directly from methods.

60. Property file name by default is:

- A. `application.yml`
- B. `application.properties`
- C. both supported
- D. `config.properties`

Answer: **C**

Boot supports both `.properties` and `.yml`; it picks them from classpath or `/config`.

61. Expose health/metrics via:

- A. `spring-boot-starter-test`
- B. Actuator
- C. DevTools
- D. Data JPA

Answer: **B**

Actuator adds production endpoints like `/actuator/health` and `/actuator/metrics`.

62. Recommended DI style in Spring Boot:

- A. Field injection
- B. Constructor injection
- C. Static injection
- D. Manual `new()`

Answer: **B**

Constructor injection is test-friendly and enforces immutability of dependencies.

63. To map GET `/users` use:

- A. `@GetMapping('/users')`
- B. `@RequestMapping(method=POST)`

- C. `@PostMapping('/users')`
- D. `@Mapping(GET)`

Answer: **A**

`@GetMapping` is a composed annotation equivalent to `@RequestMapping(method=GET)`.

64. Which interface provides CRUD + paging?

- A. `CrudRepository`
- B. `PagingAndSortingRepository`
- C. `JpaRepository`
- D. `Repository`

Answer: **C**

`JpaRepository` extends `PagingAndSortingRepository` and `CrudRepository` with JPA-specific APIs.

65. DDL auto setting to create tables each start:

- A. none
- B. validate
- C. update
- D. create-drop

Answer: **D**

`create-drop` creates schema at startup and drops it on shutdown (useful for tests).

66. To externalize config by profile use:

- A. `application-dev.properties`
- B. `dev.properties` only
- C. `profiles.yml` only
- D. `bootstrap-only`

Answer: **A**

Profile-specific files like `application-dev.properties` override defaults when 'dev' is active.

67. To map DTO to Entity using annotations minimal effort:

- A. `@Map`
- B. Use `MapStruct/ModelMapper`
- C. `@EntityMap`
- D. `@AutoMap`

Answer: **B**

Libraries like `MapStruct` generate type-safe mappers; `ModelMapper` does convention-based mapping.

68. Which annotation handles exceptions globally?

- A. `@ControllerAdvice + @ExceptionHandler`

- B. `@RestControllerAdvice` only
- C. `@Catch`
- D. `@ErrorHandler`

Answer: A

`@ControllerAdvice` allows centralized `@ExceptionHandler` methods for all controllers.

69. To run code at startup use:

- A. `CommandLineRunner`
- B. `@PostConstruct` only
- C. `ApplicationStopper`
- D. `@Startup`

Answer: A

Beans implementing `CommandLineRunner`/`ApplicationRunner` execute after the context is ready.

70. For transactions annotate service with:

- A. `@Transactional`
- B. `@Atomic`
- C. `@Synchronized`
- D. `@Tx`

Answer: A

`@Transactional` demarcates method/class boundaries for transactional behavior.

71. To validate request body fields use:

- A. `@Valid` and Bean Validation annotations
- B. Manual if-else only
- C. XML schema
- D. `@Validate`

Answer: A

Use `javax.validation` annotations (`@NotNull`, `@Size`) and trigger with `@Valid` in controller.

72. CORS for a controller can be enabled by:

- A. `@CrossOrigin`
- B. `@CORS`
- C. `@AllowOrigin`
- D. `server.cors=true`

Answer: A

Annotate controller/method with `@CrossOrigin` or configure global `CorsRegistry`.

73. H2 in-memory DB dependency:

- A. spring-boot-starter-h2
- B. com.h2database:h2
- C. spring-boot-h2-starter
- D. org.h2:h2boot

Answer: **B**

Add the H2 JDBC driver artifact 'com.h2database:h2'; Boot autoconfigures console if enabled.

74. Logging framework default in Spring Boot:

- A. Log4j
- B. java.util.logging
- C. Logback via SLF4J
- D. System.out

Answer: **C**

Spring Boot uses Logback (through SLF4J facade) by default.

75. To change server port use property:

- A. server.http.port
- B. server.port
- C. spring.server.port
- D. server.tomcat.port

Answer: **B**

server.port=8081 changes the embedded server's port.

76. To create REST docs automatically you can use:

- A. Swagger/OpenAPI (springdoc-openapi)
- B. Spring Boot Docs Starter
- C. Javadoc only
- D. JUnit

Answer: **A**

OpenAPI docs + Swagger UI provide interactive API docs from annotations.

77. Which annotation binds path variable in URL?

- A. @PathParam
- B. @PathVariable
- C. @QueryParam
- D. @Param

Answer: **B**

@PathVariable maps the {id} segment in the URI to a method parameter.

78. Which annotation binds query parameter?

- A. @RequestParam
- B. @Query
- C. @Param
- D. @UrlParam

Answer: A

@RequestParam reads ?key=value pairs from the URL or form data.

79. To schedule tasks use:

- A. @EnableScheduling + @Scheduled
- B. @CronOnly
- C. @Timer
- D. @ScheduleIt

Answer: A

Enable scheduling at config class and annotate methods with @Scheduled(cron=...).

80. Primary benefit of Spring Boot auto-configuration:

- A. Disables configuration
- B. Reduces manual setup based on classpath & properties
- C. Forces XML
- D. Removes profiles

Answer: B

Auto-config inspects the classpath and properties to wire sensible defaults, minimizing boilerplate.

81. Angular is primarily written in:

- A. JavaScript
- B. TypeScript
- C. Dart
- D. Java

Answer: B

Angular uses TypeScript (a superset of JavaScript) for static typing and modern language features.

82. CLI command to generate a component:

- A. ng g c
- B. ng new component
- C. ng create component
- D. ng c

Answer: A

The shorthand 'ng g c' invokes 'ng generate component'.

83. Two-way binding syntax is:

- A. {}{}
- B. []
- C. ()
- D. [()]

Answer: **D**

Banana-in-a-box [()] combines property and event binding for ngModel.

84. Which decorator defines a component?

- A. @Injectable
- B. @Directive
- C. @Component
- D. @NgModule

Answer: **C**

@Component decorates a class with selector, template, and styles metadata.

85. Service registration at root is done by:

- A. providedIn: 'root'
- B. imports: [RootService]
- C. bootstrap: [Service]
- D. declarations: [Service]

Answer: **A**

providedIn: 'root' makes a tree-shakable singleton service across the app.

86. Which is NOT a structural directive?

- A. \*ngIf
- B. \*ngFor
- C. \*ngSwitchCase
- D. [ngClass]

Answer: **D**

[ngClass] is an attribute directive (changes styling), not structural (which alters the DOM tree).

87. Routing configuration is defined using:

- A. Routes array and RouterModule.forRoot()
- B. AppRoutingService
- C. RouteProvider
- D. RouteConfig.json

Answer: **A**

Define Routes[] and import RouterModule.forRoot(routes) in the root module.

88. HttpClient belongs to which module?

- A. @angular/forms
- B. @angular/common/http
- C. @angular/router
- D. @angular/platform-browser

Answer: **B**

Import HttpClientModule from @angular/common/http to use HttpClient.

89. Which pipe formats dates?

- A. number
- B. json
- C. date
- D. currency

Answer: **C**

The built-in 'date' pipe formats Date objects or ISO strings.

90. To share data between unrelated components you use:

- A. Input/Output only
- B. A shared service with observables
- C. Template variables
- D. ngModel

Answer: **B**

A singleton service with Subject/BehaviorSubject allows pub-sub style communication.

91. Ahead-of-Time compilation acronym:

- A. JIT
- B. AOT
- C. Ivy
- D. SSR

Answer: **B**

AOT compiles templates at build time for faster startup and earlier error detection.

92. Which command builds a production bundle?

- A. ng build --prod
- B. ng serve -p
- C. ng make prod

D. ng deploy --prod

Answer: A

'ng build --prod' (or 'ng build --configuration production') produces optimized bundles.

93. Which lifecycle hook is called once after first ngOnChanges?

- A. ngOnInit
- B. ngAfterViewInit
- C. ngDoCheck
- D. ngOnDestroy

Answer: A

ngOnInit is ideal for initializing component state after inputs are set.

94. Reactive Forms are created using:

- A. FormGroup, FormControl, FormBuilder
- B. NgModel only
- C. Template variables
- D. DOM APIs

Answer: A

Reactive forms are model-driven using FormGroup/FormControl and optionally FormBuilder.

95. Which operator is commonly used to chain HTTP calls?

- A. map
- B. mergeMap/switchMap
- C. delay
- D. catchError only

Answer: B

mergeMap/switchMap flatten inner Observables created from HTTP calls (e.g., dependent requests).

96. To protect a route you use:

- A. RouteGuard
- B. CanActivate guard
- C. RouterProtect
- D. SecureRoute

Answer: B

Implement CanActivate (and others like CanLoad) to control navigation.

97. Change detection strategy to optimize is:

- A. Default
- B. OnPush

- C. Manual
- D. Optimized

Answer: **B**

OnPush checks only when inputs change or events fire, reducing work for large trees.

98. Which file lists project dependencies?

- A. angular.json
- B. package.json
- C. tsconfig.json
- D. polyfills.ts

Answer: **B**

package.json lists npm dependencies; angular.json configures CLI/project settings.

99. To emit events from child to parent use:

- A. @Input
- B. @Output + EventEmitter
- C. @EmitOnly
- D. @Event

Answer: **B**

`@Output()` event = new EventEmitter(); then event.emit(value) and (event) binding in parent.

100. Which module is required for ngModel?

- A. ReactiveFormsModule
- B. FormsModule
- C. HttpClientModule
- D. RouterModule

Answer: **B**

Template-driven forms (ngModel) need FormsModule; ReactiveFormsModule is for reactive forms.