Practice Questions

✔️ **Section A: Spring Boot & REST API (15 Questions)**

1. **What is Spring Boot?**
   A framework that simplifies Spring development using auto-configuration, embedded servers, and starter dependencies.

2. **What does @SpringBootApplication do?**
   Combines @Configuration, @EnableAutoConfiguration, and @ComponentScan.

3. **What is auto-configuration in Spring Boot?**
   It automatically configures beans based on classpath and properties.

4. **Difference between spring-boot-starter-web and spring-boot-starter-webmvc?**
   starter-web is standard for web apps; starter-webmvc is rarely needed separately.

5. **What is the default embedded server in Spring Boot?**
   Tomcat.

6. **What is the role of application.properties?**
   To configure application settings like port, DB credentials, etc.

7. **Explain @RestController vs @Controller.**
   @RestController returns JSON/XML directly; @Controller returns views.

8. **What is ResponseEntity used for?**
   To customize HTTP status codes and headers in responses.

9. **Explain @RequestMapping vs @GetMapping.**
   @RequestMapping is generic; @GetMapping is specialized for GET requests.

10. **What is the default port for Spring Boot?**
    8080.

11. **How do you enable CORS in Spring Boot?**
    Using @CrossOrigin or global configuration.

12. **What is the difference between @PathVariable and @RequestParam?**
    @PathVariable extracts values from URI; @RequestParam extracts query parameters.

13. **How do you handle PATCH requests in Spring Boot?**
    Using @PatchMapping annotation.

14. **What is the role of @Validated?**
    To enable validation on method parameters.

Practice Questions

15. **How do you start a Spring Boot application?**
    Using SpringApplication.run() or mvn spring-boot:run.

✔️ **Section B: Core Java (15 Questions)**

16. **Difference between == and equals()?**
    == compares references; equals() compares content.

17. **What is the difference between StringBuilder and StringBuffer?**
    StringBuilder is faster but not thread-safe; StringBuffer is synchronized.

18. **Explain HashMap vs ConcurrentHashMap.**
    HashMap is not thread-safe; ConcurrentHashMap allows concurrent access.

19. **What is the difference between final, finally, and finalize()?**
    final keyword, finally block in exceptions, finalize() method for GC.

20. **What is the difference between ArrayList and LinkedList?**
    ArrayList is faster for random access; LinkedList is better for insert/delete.

21. **Explain Java Streams.**
    Functional-style operations on collections for concise and parallel processing.

22. **What is the difference between throw and throws?**
    throw is used to throw an exception; throws declare exceptions in method signature.

23. **What is the difference between checked and unchecked exceptions?**
    Checked must be handled or declared; unchecked are runtime exceptions.

24. **What is the difference between volatile and synchronized?**
    volatile ensures visibility; synchronized ensures atomicity and visibility.

25. **What is the difference between abstract class and interface?**
    Abstract class can have state; interface is purely contract-based.

26. **Explain Optional in Java.**
    A container to avoid NullPointerException by representing optional values.

27. **What is the difference between HashSet and TreeSet?**
    HashSet is unordered; TreeSet is sorted.

28. **Explain immutability in Java.**
    Immutable objects cannot be changed after creation (e.g., String).

29. **What is the difference between ExecutorService and Thread?**
    ExecutorService manages thread pools; Thread is a single execution unit.

30. **What is the difference between static and instance methods?**
    Static belongs to class; instance belongs to object.

✔️ **Section C: Hibernate / JPA (10 Questions)**

31. **What is Hibernate?**
    An ORM framework mapping Java objects to database tables.

32. **Difference between get() and load()?**
    get() returns null if not found; load() throws exception and uses lazy loading.

33. **What is the role of @Entity?**
    Marks a class as a JPA entity.

34. **Explain FetchType.LAZY vs FetchType.EAGER.**
    LAZY loads related entities on demand; EAGER loads immediately.

35. **What is the difference between save() and persist()?**
    save() returns ID; persist() does not return anything.

36. **What is the role of @JoinColumn?**
    Specifies foreign key column in relationships.

37. **What is the difference between @OneToMany and @ManyToOne?**
    @OneToMany defines parent-child; @ManyToOne defines child-parent.

38. **What is the role of @Enumerated(EnumType.STRING)?**
    Stores enum as string in DB instead of ordinal.

39. **What is the difference between merge() and update()?**
    merge() copies state to a managed entity; update() reattaches a detached entity.

40. **Explain cascading in JPA.**
    Cascade operations propagate from parent to child entities (e.g., CascadeType.ALL).

✔️ **Section D: Validation & Exception Handling (5 Questions)**

41. **What does @Valid do?**
    Triggers Bean Validation on request bodies.

42. **Difference between @Valid and @Validated?**
    @Valid works on method parameters; @Validated supports groups.

43. **What is @RestControllerAdvice used for?**
    Global exception handling across controllers.

Practice Questions

44. **What is ConstraintViolationException?**
    Occurs when validation on request parameters fails.

45. **How do you return custom error responses?**
    Using @ExceptionHandler and building a response map or DTO.

✔️ **Section E: Swagger / OpenAPI (5 Questions)**

46. **What is OpenAPI Specification?**
    A standard for describing REST APIs in JSON/YAML.

47. **What is the default Swagger UI URL in Springdoc?**
    /swagger-ui.html or /swagger-ui/index.html.

48. **How do you customize API documentation title?**
    Using @OpenAPIDefinition(info = @Info(...)).

49. **Difference between Swagger 2 and OpenAPI 3?**
    OpenAPI 3 supports better schema definitions and callbacks.

50. **How do you expose API docs in JSON format?**
    Springdoc provides /v3/api-docs endpoint.

MCQ Questions

✔️ **Section A: Spring Boot & REST API (15 MCQs)**

1. **Which annotation combines @Configuration, @EnableAutoConfiguration, and @ComponentScan?**
   a) @SpringBootTest
   b) @SpringBootApplication ✔️
   c) @EnableWebMvc
   d) @RestController

2. **Default embedded server in Spring Boot?**
   a) Jetty
   b) Tomcat ✔️
   c) Undertow
   d) Netty

3. **Which starter is standard for web apps?**
   a) spring-boot-starter-webmvc
   b) spring-boot-starter-web ✔️
   c) spring-boot-starter-rest
   d) spring-boot-starter-core

4. **Default port for Spring Boot?**
   a) 80
   b) 8080 ✓
   c) 8081
   d) 9090

5. **Which annotation returns JSON directly?**
   a) @Controller
   b) @RestController ✓
   c) @ResponseBody
   d) @RequestMapping

6. **Which annotation maps GET requests?**
   a) @RequestMapping
   b) @GetMapping ✓
   c) @PostMapping
   d) @PatchMapping

7. **Which class starts the Spring Boot app?**
   a) SpringApplication ✓
   b) ApplicationContext
   c) DispatcherServlet
   d) ServletInitializer

8. **Which annotation enables global exception handling?**
   a) @ControllerAdvice
   b) @RestControllerAdvice ✓
   c) @ExceptionHandler
   d) @Advice

9. **Which annotation validates request body?**
   a) @Valid ✓
   b) @Validated
   c) @Check
   d) @Assert

10. **Which annotation is used for PATCH requests?**
    a) @PatchMapping ✓
    b) @PutMapping
    c) @UpdateMapping
    d) @ModifyMapping

11. **Which property sets server port?**
    a) server.port ✓

b) spring.port

c) server.http.port

d) spring.server.port

12. **Which annotation is used for query parameters?**

a) @PathVariable

b) @RequestParam ✓

c) @QueryParam

d) @Param

13. **Which annotation is used for path variables?**

a) @RequestParam

b) @PathVariable ✓

c) @Variable

d) @Param

14. **Which annotation enables CORS?**

a) @CrossOrigin ✓

b) @EnableCors

c) @AllowOrigin

d) @CorsMapping

15. **Which annotation is used for API documentation metadata?**

a) @Api

b) @OpenAPIDefinition ✓

c) @SwaggerDefinition

d) @ApiDoc

✔ **Section B: Core Java (15 MCQs)**

16. **Which compares object content?**

a) ==

b) equals() ✓

c) compareTo()

d) hashCode()

17. **Which is thread-safe?**

a) StringBuilder

b) StringBuffer ✓

c) String

d) StringJoiner

Practice Questions

18. **Which collection is unordered?**

    a) HashSet ✓

    b) TreeSet

    c) LinkedHashSet

    d) PriorityQueue

19. **Which keyword prevents inheritance?**

    a) finally

    b) final ✓

    c) static

    d) const

20. **Which block always executes?**

    a) catch

    b) finally ✓

    c) throw

    d) throws

21. **Which interface supports functional programming?**

    a) Runnable

    b) Stream ✓

    c) Callable

    d) Comparator

22. **Which exception must be handled?**

    a) Checked ✓

    b) Unchecked

    c) Runtime

    d) Error

23. **Which ensures visibility across threads?**

    a) volatile ✓

    b) synchronized

    c) transient

    d) static

24. **Which is immutable?**

    a) String ✓

    b) StringBuilder

    c) StringBuffer

    d) CharSequence

25. **Which collection maintains insertion order?**

    a) HashSet

b) LinkedHashSet ✓
c) TreeSet
d) HashMap

26. **Which method is called before GC destroys object?**

a) finalize() ✓
b) finally()
c) dispose()
d) destroy()

27. **Which class avoids NullPointerException?**

a) Optional ✓
b) Nullable
c) SafeObject
d) Wrapper

28. **Which collection is sorted?**

a) TreeSet ✓
b) HashSet
c) LinkedHashSet
d) ArrayList

29. **Which manages thread pools?**

a) Thread
b) ExecutorService ✓
c) Runnable
d) Callable

30. **Which keyword is used for inheritance?**

a) extends ✓
b) implements
c) inherit
d) super

✔️ **Section C: Hibernate / JPA (10 MCQs)**

31. **Which annotation marks a JPA entity?**

a) @Table
b) @Entity ✓
c) @Column
d) @Persistence

32. **Which fetch type loads lazily?**

    a) EAGER

    b) LAZY ✔

    c) DEFAULT

    d) DEFERRED

33. **Which annotation defines foreign key column?**

    a) @JoinColumn ✔

    b) @Column

    c) @ForeignKey

    d) @Key

34. **Which annotation maps enum as string?**

    a) @Enumerated(EnumType.STRING) ✔

    b) @EnumString

    c) @EnumType

    d) @StringEnum

35. **Which method returns null if entity not found?**

    a) load()

    b) get() ✔

    c) find()

    d) fetch()

36. **Which annotation defines one-to-many relationship?**

    a) @ManyToOne

    b) @OneToMany ✔

    c) @OneToOne

    d) @ManyToMany

37. **Which annotation defines primary key?**

    a) @Id ✔

    b) @PrimaryKey

    c) @Key

    d) @PK

38. **Which annotation defines table name?**

    a) @Entity

    b) @Table ✔

    c) @Column

    d) @JoinTable

39. **Which cascade type propagates all operations?**

    a) CascadeType.ALL ✔

b) CascadeType.MERGE

c) CascadeType.PERSIST

d) CascadeType.REMOVE

40. **Which annotation defines column name?**

a) @Column ✔

b) @Field

c) @JoinColumn

d) @Attribute

✔ **Section D: Validation & Exception Handling (5 MCQs)**

41. **Which annotation triggers validation?**

a) @Valid ✔

b) @Validated

c) @Check

d) @Assert

42. **Which annotation supports validation groups?**

a) @Valid

b) @Validated ✔

c) @GroupCheck

d) @AssertGroup

43. **Which annotation handles global exceptions?**

a) @RestControllerAdvice ✔

b) @ControllerAdvice

c) @ExceptionHandler

d) @Advice

44. **Which exception occurs on invalid request params?**

a) ConstraintViolationException ✔

b) MethodArgumentNotValidException

c) IllegalArgumentException

d) ValidationException

45. **Which annotation maps custom exception handler?**

a) @ExceptionHandler ✔

b) @ErrorHandler

c) @Catch

d) @HandleException

Practice Questions

✔ Section E: Swagger / OpenAPI (5 MCQs)

46. **Which dependency adds Swagger UI in Spring Boot?**
    a) springdoc-openapi-starter-webmvc-ui ✓
    b) swagger-spring-boot-starter
    c) spring-swagger-ui
    d) openapi-ui

47. **Default Swagger UI URL?**
    a) /swagger-ui.html ✓
    b) /swagger.html
    c) /api-docs
    d) /swagger

48. **Which annotation customizes API info?**
    a) @OpenAPIDefinition ✓
    b) @ApiInfo
    c) @SwaggerDefinition
    d) @ApiDoc

49. **Which endpoint returns OpenAPI JSON?**
    a) /v3/api-docs ✓
    b) /swagger.json
    c) /api-docs
    d) /openapi

50. **Which version introduced OpenAPI 3?**
    a) Swagger 2
    b) OpenAPI 3 ✓
    c) Swagger 1
    d) OpenAPI 2