Q1. Which of the following is NOT a key feature of ASP.NET Core?

a. Cross-platform support

b. Lightweight and high-performance

[✓ c. Legacy code compatibility]

d. Built-in dependency injection

Q2. What is the primary purpose of the Startup.cs file in an ASP.NET Core application?

[✓ a. To configure middleware and services]

b. To define the models for the application

c. To handle HTTP requests and responses

d. To manage the application's UI components

Q3. Which of the following is a built-in dependency injection container in ASP.NET Core?

a. Autofac

b. Unity

c. StructureMap

[✓ d. ConfigureServices]

Q4. What are the default hosting environments for an ASP.NET Core application?

a. Development

b. Production

c. Staging

d. Local

[✓ A,B,C is correct]

Q5. In ASP.NET Core, what is the purpose of the wwwroot folder?

[✓ a. To store static files and assets]

b. To store configuration files

c. To store server-side logic

d. To store middleware components

Q6. Which of the following is used to define routes in ASP.NET Core?

a. RouteConfig.cs

[✓ b. Startup.cs]

c. RouteConfig.json

d. ConfigureServices.cs

Q7. Which of the following is a correct way to handle errors in ASP.NET Core?

a. Use try-catch blocks in controllers

b. Use the ErrorController class

[✓ c. Use the global exception filter]

d. Use the HandleErrorAttribute

Q8. What does CORS stand for in the context of ASP.NET Core?

[✓ a. Cross-Origin Resource Sharing]

b. Cross-Origin Request Sharing

c. Cross-Origin Resource System

d. Cross-Origin Request System

Q9. What is the purpose of the ConfigureServices method in the Startup class in ASP.NET Core?

[✓ a. To configure services used by the application]

b. To configure the application's routes

c. To handle HTTP requests

d. To define the application's models

In ASP.NET Core, what is the purpose of the "appsettings.json" file?
To store application configuration settings

What is the role of the "wwwroot" folder in an ASP.NET Core project?
To serve static web content

Which of the following authentication mechanisms is commonly used in ASP.NET Core to secure web applications?
JWT (JSON Web Tokens)

What is the purpose of the Entity Framework in ASP.NET Core?
To provide data access and ORM capabilities

Razor Pages in ASP.NET Core?
Razor Pages combine HTML markup and C# code for creating web pages.

In ASP.NET Core, what is the purpose of the "Middleware"?
To process HTTP requests and responses in a pipeline

Which of the following HTTP status codes typically represents a successful response in ASP.NET Core?
200 (OK)

What is the role of the ConfigureServices method in the Startup class of an ASP.NET Core application?
To register and configure services used by the application

Which of the following is NOT a valid data type in C# and ASP.NET Core?
array

MCQS

Question 1:

What is Spring Boot?

[✓ a) A lightweight Java application framework]

b) A Java virtual machine

c) A version control system

d) An integrated development environment

Question 2:

In Spring Boot, what is the primary purpose of the @SpringBootApplication annotation?

[✓ a) Marking the application as a Spring Boot application] configuring the application with sensible defaults

b) Enabling database connectivity

c) Defining a RESTful API

d) Configuring security settings

Question 3:

What is the default embedded web server used by Spring Boot?

[✓ a) Apache Tomcat] – without external server configuration

b) Jetty

c) WebSphere

d) JBoss

Question 4:

Which file is used for configuring application properties in Spring Boot?

a) application.yml

[✓ b) application.properties]

c) config.xml

d) settings.properties

Question 5:

How does Spring Boot simplify dependency management?

[✓ a) By automatically managing version conflicts]

b) By removing the need for dependencies

c) By enforcing specific dependency versions

d) By providing a separate dependency management tool

Question 6:

Which annotation in Spring Boot is used to define a RESTful endpoint?

[✓ a) @RestController]

b) @Controller

c) @RequestMapping

d) @Service

Question 7:

Which Spring Boot feature simplifies the creation of CRUD (Create, Read, Update, Delete) operations for repositories?

[✓ a) Spring Data JPA]

b) Spring MVC

c) Spring Boot Starter

d) Spring AOP

Question 8:

Which annotation in Spring Boot is used for automatic component scanning and bean registration?

[✓ a) @ComponentScan]

b) @Bean

c) @Autowired

d) @Configuration

Question 9:

What is the purpose of the @SpringBootTest annotation in Spring Boot?

[✓ a) It marks a class as a Spring Boot test class]

b) It configures the application properties for testing

c) It defines the main application class

d) It is used for handling exceptions in tests

Q 10 - Which of the following is true about Maven Conventions?

[✓ A - Maven uses Convention over Configuration which means developers are not required to create build process themselves.]

[✓ B - Developers using maven do not have to mention each and every configuration details.]

C - Both of the above.

D - None of the above.

Q11. Which annotation is used to inject a property value from a properties file into a Spring Boot component?

properties file @value

property value → component

a. @InjectValue

b. @PropertyInject

c. @Autowired

[✓ d. @Value]

Q12. Which annotation is used to enable transaction management in Spring Boot?

a. @Transactional

[✓ b. @EnableTransactionManagement]

c. @TransactionEnabled

d. @EnableTransactions

Q13. What is the purpose of the @Repository annotation in Spring Boot?

a. It marks a class as a Spring Bean.

b. It defines a REST endpoint.

c. It specifies the database connection details.

@Repository → DAO

Data Access Object

[✓ d. It indicates that a class is a Data Access Object (DAO).]

Q14. Which HTTP methods are commonly used in RESTful APIs for performing CRUD operations?

a. GET, POST, DELETE, UPDATE

b. CREATE, READ, MODIFY, DELETE

[✓ c. GET, POST, PUT, DELETE]

d. FETCH, INSERT, ALTER, REMOVE

Q15. What is Hibernate in the context of Spring Boot?

a. A programming language

b. A relational database management system

[✓ c. An Object-Relational Mapping (ORM) framework]

d. A web application framework

Q16. In a Spring Boot CRUD API, what does the @GetMapping annotation indicate in a controller method?

[✓ a. It maps a method to handle HTTP GET requests.]

b. It defines the entity model.

c. It specifies the database table.

d. It enables cross-origin requests.

Q17. In a Spring Boot CRUD API, what does the following code snippet do?

```
@PostMapping("/products")
public ResponseEntity<Product> createProduct(@RequestBody Product product) {
   // Method logic
}
```

a. Retrieves a specific product by its ID.

b. Updates an existing product in the database.

c. Deletes a product from the database.

[✓ d. Creates a new product with the data provided in the request body.]

Q18. In a Spring Boot and Hibernate application, what is the primary role of the EntityManager when used in conjunction with Hibernate?

a. It handles HTTP requests and responses.

b. It manages the persistence context and performs database operations.

c. It defines RESTful endpoints.

d. It configures the data source.

Q19. In a Spring Boot application using Hibernate, what is the purpose of the @Entity annotation when applied to a class?

a. It specifies the primary key of the entity.

b. It defines the entity's table name in the database.

c. It marks the class as a persistent entity.

d. It maps the class to a RESTful endpoint.

Q20.In a Spring Boot application, what is the primary purpose of the @RequestMapping annotation when applied to a controller class?

a. It defines the primary key for the class.

b. It specifies the HTTP methods allowed for the class.

c. It marks the class as a repository.

d. It defines the base URL for all methods in the class.

SCENARIO BASED MCQS

Scenario 1

Imagine you are developing a Spring Boot application for a bookstore. The application will manage books, authors, and customer orders.

Question 1:

You need to create a RESTful API endpoint to retrieve book details by their ISBN (International Standard Book Number). What annotation should you use to map this API endpoint?

[✓ a) @GetMapping] Map an HTTP GET request to a specific method

b) @PostMapping

c) @RequestMapping

d) @GetMapping("/books")

Question 2:

In the bookstore application, you want to ensure that the order information is saved to the database. Which Spring Boot feature would you use for easy database interaction and persistence?

[✓ a) Spring Data JPA] → Simplifies the database interaction and persistance

b) Spring Boot Security

c) Spring AOP

d) Spring Boot Actuator

Question 3:

For the bookstore application, you need to handle authentication and authorization. Which annotation is typically used for securing RESTful endpoints in Spring Boot?

a) @Secure

b) @Authorize

[✓ c) @Secured] → Securing RESTful endpoints

specify which roles are allowed to access specific methods

d) @Security

Question 4:

To manage book information, you want to create a form where users can add new books. Which annotation should be used in the Spring Boot controller to handle the form submission?

[✓ a) @PostMapping] handle form submissions and create new resources

b) @GetMapping

c) @RequestMapping

d) @PutMapping

Question 5:

In the bookstore application, you need to retrieve a list of authors associated with a particular book. Which Spring Boot annotation is commonly used to handle requests that involve multiple entities?

[✓ a) @GetMapping]

b) @RequestMapping

c) @PathVariable

d) @RequestParam

Scenario 2:

You are developing a Spring Boot application for a library management system. The application will manage information about books, authors, and user borrowing records using Hibernate for database persistence.

Question 1:

In the library management system, which annotation would you use in Hibernate to map a Java class to a database table?

[✓ a) @Entity]

b) @Table

c) @Column

d) @Id

Question 2:

For the "Book" entity in the library management system, you want to define the primary key. Which Hibernate annotation would you use to achieve this?

[✓ a) @Id]

b) @PrimaryKey

c) @GeneratedValue

d) @EntityId

Question 3:

In the library management system, you need to map a Java class property to a specific column in the database table. Which Hibernate annotation would you use for this purpose?

[✓ a) @Column]

b) @Table

c) @Entity

d) @JoinColumn

Question 4:

To define a one-to-many relationship between the "Author" and "Book" entities in the library management system, which Hibernate annotation would you use?

[✓ a) @OneToMany]

b) @ManyToOne

c) @OneToOne

d) @ManyToMany

Question 5:

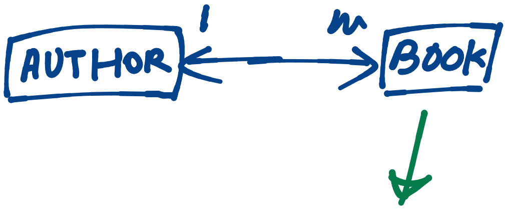
In the library management system, you want to eagerly fetch the associated authors whenever you retrieve a book. Which Hibernate annotation would you use to achieve this?

[✓ a) @ManyToOne(fetch = FetchType.EAGER)]

b) @OneToMany(fetch = FetchType.LAZY)

c) @ManyToOne(fetch = FetchType.LAZY)

d) @OneToMany(fetch = FetchType.EAGER)



Scenario 3 :

You are developing a Spring Boot application for an online shopping platform. The application will handle various aspects of product management, customer orders, and payments. Spring's Dependency Injection and Inversion of Control (IoC) are essential concepts used extensively in this application.

Question 1:

In the online shopping platform application, what is the main purpose of Dependency Injection?

a) To handle HTTP requests and responses

b) To manage and provide application components

c) To define database schemas

d) To establish network connections

Dependency injection → primarily used to manage and provide application components, making them available

Question 2:

In the context of the online shopping platform application, what does IoC (Inversion of Control) mean in Spring?

a) Spring managing the application's lifecycle

b) Allowing the developer to control application components

c) Allowing the container to manage the components and their dependencies

d) Inverting the control flow of the application

IOC allowing the spring container to mange the components and their dependencies, rather than the developer controlling them explicitly

Question 3:

In Spring, what is the role of a Bean in the context of Dependency Injection and IoC?

a) A coffee bean that Spring uses for dependency management

b) A Java class managed by the Spring container

c) A design pattern used for dependency management

d) A database entry representing a class

Bean refers to a Java class manged by the spring Container, which can be injected in to other components

Question 4:

In the online shopping platform application, you want to inject an instance of a "PaymentService" into your "OrderService" class. Which Spring annotation would you use to achieve this?

[✓ a) @Autowired]

b) @Inject

c) @Component

d) @Service

Question 5:

In the online shopping platform application, you need to ensure that only a single instance of a particular service is created and used throughout the application. Which Spring annotation would you use to achieve this?

[✓ a) @Scope("singleton")] ← Design pattern

b) @Scope("prototype")

c) @Component

d) @Service