



Higher Diploma in Software Engineering

Model Paper

Advanced API Development (ITS 1114)

Part A – Multiple Choice Questions (40 × 1 = 40 Marks)

1. Which HTTP method is typically idempotent?
 - a) POST
 - b) PUT
 - c) PATCH
 - d) CONNECT
2. The role of a Web Container in Java EE is to:
 - a) Manage low-level file I/O operations
 - b) Provide servlet lifecycle management
 - c) Generate dynamic HTML from JSP only
 - d) Act as a database connection pool
3. In RESTful APIs, which format is most commonly used for data exchange?
 - a) XML
 - b) JSON
 - c) CSV
 - d) HTML
4. What does CORS stand for?
 - a) Cross-Origin Resource Sharing
 - b) Centralized Object Runtime Specification
 - c) Client Origin Request System
 - d) Cross-Object Response Service
5. Which annotation is used to define a Spring Bean?
 - a) @Autowired
 - b) @Bean
 - c) @Component
 - d) @Configuration
6. IoC in Spring Framework primarily means:
 - a) Inheritance of Classes
 - b) Inversion of Control
 - c) Injection of Components
 - d) Internal Object Communication
7. Which of the following is **not** a build tool?
 - a) Maven
 - b) Gradle
 - c) Ant

d) Hibernate

8. Which layer is introduced when using Spring Data JPA?

- a) Persistence Layer
- b) Presentation Layer
- c) Service Layer
- d) Business Layer

9. In Hibernate, the annotation `@Entity` is used to:

- a) Define a database table schema
- b) Map a class to a database table
- c) Configure dependency injection
- d) Handle JSON serialization

10. Spring Boot's `application.properties` is used for:

- a) Database schema design
- b) Dependency injection
- c) Centralized configuration
- d) Mapping URL routes

11. Which dependency manager does Spring Boot internally rely on?

- a) Ant
- b) Gradle
- c) Maven
- d) Ivy

12. The default embedded server in Spring Boot is:

- a) Jetty
- b) Undertow
- c) Tomcat
- d) Glassfish

13. Which HTTP status code represents **Unauthorized** access?

- a) 400
- b) 401
- c) 403
- d) 404

14. MongoDB is classified as:

- a) Relational DB
- b) Columnar DB
- c) NoSQL DB
- d) Graph DB

15. Which annotation is used for **request mapping** in Spring?
- a) @Controller
 - b) @RequestParam
 - c) @RequestMapping
 - d) @Autowired
16. Dependency Injection in Spring can be achieved by:
- a) Constructor Injection
 - b) Setter Injection
 - c) Field Injection
 - d) All of the above
17. Which of the following is a **stateless authentication mechanism**?
- a) Session-based authentication
 - b) JWT
 - c) Cookie-based authentication
 - d) Basic Authentication with server state
18. OAuth 2.0 provides:
- a) Authorization framework
 - b) Authentication only
 - c) Database connectivity
 - d) API caching
19. In Spring Boot, the annotation @SpringBootApplication is a combination of:
- a) @Configuration + @ComponentScan + @EnableAutoConfiguration
 - b) @Service + @Bean + @EnableJpaRepositories
 - c) @Repository + @Autowired + @RestController
 - d) None of the above
20. Which annotation is used for defining **custom exception handlers** in Spring?
- a) @RestControllerAdvice
 - b) @ExceptionHandler
 - c) Both a and b
 - d) @Controller
21. What is the main benefit of @Autowired in Spring?
- a) Automatic bean wiring
 - b) JSON parsing
 - c) HTTP request mapping
 - d) Database indexing
22. Which HTTP status code represents **Resource Not Found**?
- a) 200

- b) 201
- c) 404
- d) 500

23. Which Spring Boot module helps in managing logs?

- a) Spring Boot Actuator
- b) Spring Boot Logging
- c) SLF4J with Logback
- d) Spring Boot Admin

24. Which of the following is true about JSON?

- a) It is a binary data format
- b) Supports nested objects
- c) Cannot be used in APIs
- d) XML is faster than JSON

25. Which JPA annotation specifies a **primary key**?

- a) @GeneratedValue
- b) @Id
- c) @Column
- d) @Table

26. In Spring Security, @EnableWebSecurity is used to:

- a) Enable database connections
- b) Configure authentication & authorization
- c) Manage logging
- d) Build API documentation

27. JWTs are composed of:

- a) Header, Payload, Signature
- b) Request, Response, Token
- c) Key, Value, Hash
- d) Client, Server, Middleware

28. Which HTTP method is **not** idempotent?

- a) GET
- b) PUT
- c) POST
- d) DELETE

29. Which ORM concept ensures object-table mapping?

- a) Dependency Injection
- b) Object-Relational Mapping
- c) Cross-Origin Policy

d) MVC

30. In REST API, **statelessness** means:

- a) Server does not store client state between requests
- b) API cannot handle sessions
- c) Database connection is always closed
- d) API only works with GET methods

31. Spring Boot starter dependencies simplify:

- a) Manual library imports
- b) Automatic configuration
- c) Both a and b
- d) None

32. The primary use of JNDI in API development is:

- a) File mapping
- b) Naming and directory services
- c) Logging
- d) JSON parsing

33. What does `@RestController` annotation combine?

- a) `@Controller` + `@ResponseBody`
- b) `@Service` + `@Bean`
- c) `@Repository` + `@Table`
- d) `@Component` + `@Configuration`

34. Hibernate uses which file for XML-based configuration?

- a) hibernate.xml
- b) persistence.xml
- c) hibernate.cfg.xml
- d) orm.xml

35. Spring Boot Actuator provides:

- a) Monitoring & management endpoints
- b) Database schema management
- c) Security only
- d) Object mapping

36. Which tool is used to test APIs manually?

- a) Eclipse
- b) Postman
- c) Docker
- d) Jenkins

37. In microservices, which Spring Cloud component handles **service discovery**?
- a) Eureka
 - b) Ribbon
 - c) Zuul
 - d) Hystrix
38. A **Bean Validation** in Spring Boot uses which annotation?
- a) @Valid
 - b) @Validate
 - c) @Validator
 - d) @Verified
39. Which of the following is NOT a feature of Spring Boot?
- a) Auto-configuration
 - b) Embedded servers
 - c) Externalized configuration
 - d) Manual XML configuration only
40. OAuth 2.0's **Refresh Token** is primarily used for:
- a) Getting a new access token without user login
 - b) Storing client sessions
 - c) Authenticating APIs directly
 - d) Logging exceptions

Updated Essay Section – Advanced API Development (ITS 1114)

Part B – Essay Questions (6 × 10 = 60 Marks)

Q1. Explain the architecture of the Spring Framework. Discuss the roles of IoC, AOP, and Bean management with relevant examples.

- a) Define Inversion of Control (IoC) and explain how it is achieved in Spring.
 - b) Illustrate how Bean lifecycle management works in Spring with a simple example.
 - c) Explain Aspect-Oriented Programming (AOP) with a real-world scenario.
 - d) Discuss how Spring Framework architecture supports layered application development.
-

Q2. Compare and contrast REST API and SOAP API. Highlight their advantages, disadvantages, and real-world use cases.

- a) Define REST and SOAP and explain their communication styles.
- b) Compare the data formats supported in REST vs SOAP.

- c) List the advantages and limitations of REST compared to SOAP.
 - d) Provide two industry use cases where REST is preferred and two where SOAP is preferred.
-

Q3. Describe the process of database integration in Spring Boot. Explain with examples using JPA and Hibernate.

- a) Explain the role of Spring Data JPA in persistence layer development.
 - b) Demonstrate how entities are mapped to database tables with JPA annotations.
 - c) Describe how Hibernate manages object-relational mapping.
 - d) Discuss the advantages of using Spring Boot for database integration.
-

Q4. Discuss the significance of Spring Security with JWT and OAuth 2.0. Provide a practical scenario of securing an API.

- a) Explain the core features of Spring Security.
 - b) Describe the structure of a JWT and its authentication flow.
 - c) Explain how OAuth 2.0 provides secure authorization.
 - d) Apply JWT + OAuth in a practical example of securing a RESTful API.
-

Q5. Explain the importance of Spring Boot Configurations and Testing in API development.

- a) Discuss the role of `application.properties` or `application.yml` in Spring Boot projects.
 - b) Explain how environment-specific configurations are handled in Spring Boot.
 - c) Describe how Spring Boot supports unit and integration testing.
 - d) Provide an example scenario where proper configuration and testing ensure API reliability.
-

Q6. Design a mini-project outline. Assume you are tasked with building a hospital appointment booking API. Discuss architecture, tools, frameworks, and security considerations.

- a) Define the system architecture (client, server, database, and API layers).
- b) Identify the main entities and relationships for the appointment system.
- c) Propose a technology stack (frameworks, databases, tools).
- d) Describe how you would secure the system using Spring Security and JWT.