Class Test: ICT-3209 Software Engg Project Management, Time: 3 Hours, Any 5 Q, Marks: 60

Q1 *a)* You are tasked with designing a multi-module Spring Boot application consisting of separate modules for the controller (API), business logic (service), and data access (repository) layers. Describe how you would organize the project using Maven. 3

b) How would you configure the pom.xml files to handle module dependencies and ensure proper build sequencing? 3

c) Explain the key distinctions between @Controller and @RestController in Spring Boot, particularly focusing on how response handling differs. 3

d) Given a RESTful API for an e-commerce platform, evaluate how you would structure endpoints for managing product catalogues and customer orders using REST conventions. 3

Q2 *a)* Explain the purpose of the persist(), merge(), and remove() methods in the JPA EntityManager interface. 3

b) In the context of a university registration system, analyze a situation where each method would be appropriately used during the lifecycle of student entity transactions. 3

c) How does JPA facilitate object-relational mapping between Java classes and database tables? 3

d) Illustrate your explanation with an example using @Entity, @Id, and @GeneratedValue. Evaluate the benefits JPA provides in comparison to writing raw JDBC code for the same functionality. 3

Q3 *a)* What is the role of ResultSet in JDBC when interacting with a MySQL database? 3

b) Explain how methods like next(), getString(), and getInt() are used to iterate over and extract data from the result set, using a relevant code example. 3

c) Discuss the advantages of using PreparedStatement over Statement in JDBC, especially in the context of dynamic SQL execution. 3

d) Write a short code snippet that inserts student details into a MySQL students table using PreparedStatement. 3

Q4 *a)* Describe the end-to-end request processing lifecycle in Spring MVC initiated by the DispatcherServlet. 3

b) How does it coordinate with components like HandlerMapping, Controller, ModelAndView, and ViewResolver to serve a client request? 3

c) How does Spring MVC maintain a clean separation between business logic and presentation when processing web requests? 3

d) Discuss the roles of @Controller and @RequestMapping in routing, and explain how the Model object is used to pass data to the view. Provide an example involving user authentication. 3

Q5 *a)* Explain the lifecycle of an HttpSession in a Java-based web application. 3

b) How is session continuity preserved for a logged-in user, and what mechanisms are in place to handle session expiration and invalidation for improved security? 3

18 *c)* Discuss how cookies, URL rewriting, and HttpSession handle session management in Java web applications. 3

d) Evaluate each method in terms of security, performance, client dependency, and scalability. 3

Q6 *a)* In a Java EE web application, describe how a servlet acts as a controller to coordinate communication between the model (business logic) and the view (JSP). 4

17 *b)* Demonstrate the process of setting request attributes and forwarding to a JSP for rendering a response, using an example like a product detail page. 4

c) Servlets use a multithreaded model to serve multiple requests with a single instance. Discuss the problems associated with accessing shared resources without proper synchronization. Support your explanation with an example and suggest thread-safe design strategies in servlet programming. 4

Q7 12 *a)* Describe how JDBC manages communication between a Java application and a relational database. 4

b) Illustrate the steps involved in executing a SELECT query and fetching results. Include error handling with try-catch and finally blocks. 4

11 *c)* Discuss the Singleton design pattern in Java. What problem does it solve, and how does it ensure only one instance of a class is created? 4

Q8 6 *a)* How does Java handle XML data using DOM and SAX parsers? Compare both approaches with respect to memory usage, processing speed, and use cases. Provide a scenario where SAX would be preferred over DOM. 4

9(iv)(v) *b)* Write Program to i) create a TreeMap to store the mappings of student IDs to their details, ii) Check if two LinkedLists are equal. 4

2 *c)* Compare abstract classes and interfaces in terms of multiple inheritance. When would you prefer to use an abstract class and when an interface? 4

Kuldip IT22018.