|  |  |
| --- | --- |
| **[25 minutes]**  Suppose we have an airline application that contains a domain classes **FrequentFlierMilesAccount**and **Flight**  Suppose you need to develop a REST interface for the functionality for this airline application. The REST API needs to implement the following functionality:   * **createFrequentFlierMilesAccount(customerName)**  *//returns the new created frequent flier miles account class* * **getFrequentFlierMilesAccount(accountNumber)***//returns the corresponding frequent flier miles account class* * **addPoints(accountNumber, numberOfPoints)**  *//returns the corresponding frequent flier miles account class* * **withdrawPoints(accountNumber, numberOfPoints)**  *//returns the corresponding frequent flier miles account class* * **purchaseTicketWithMiles(accountNumber, flightNumber, date, amountOfPoints)**  *//returns the corresponding flight class*   Write the **signature of all methods** of the AirlineController class, including the **necessary annotations.** A signature of a method contains the following content:  ***return-type method-name (parameter-type parameter-name, …)***  An example method signature  with corresponding annotation is:  *@Getmapping(...)*  *Customer getCustomer(int customerNumber)*  **Do NOT write the body of these methods.** Follow the**best practices** we have learned in this course.  For this question it is important that you show for each method the **mapping annotation** and the **signature** of every method.   |  | | --- | |  | |
| Question 2 of 710.0 Points  **[10 minutes]**  Select all statements that are correct.   |  | | --- | |  |  |  |  |  | | --- | --- | --- | |  | A. For every client call to a Servlet a new Servlet instance is created |  | |  | B. In SpringMVC, the DispatcherServlet is the Front Controller |  | |  | C. Flash attributes are stored in the http session |  | |  | D. The HTTP POST request is idempotent |  | |  | E. The HTTP PUT request is idempotent |  | |  | F. The HTTP DELETE request is idempotent |  | |  | G. If you implement GraphQL using Spring Boot you do not need write a Front Controller class. |  | |  | H. Suppose a Server (back-end) application want to call the client (front-end) application anytime the server has new data. This can be done using REST. |  | |  | I. Suppose a Server (back-end) application want to call the client (front-end) application anytime the server has new data. This can be done using webflux. |  | |  | J. Suppose a Server (back-end) application want to call the client (front-end) application anytime the server has new data. This can be done using web sockets. |  |  |  | | --- | | Answer Key:B, C, E, F, G, I, J | |  | |
| Question 3 of 715.0 Points  **[20 minutes]**  Suppose we have a SpringMVC web application (that shows webpages in the browser) where the **customersPage.html** shows a list of customers. The application allows the user to add a new customer to the list. Write the required code of the controller class that handles the request to add a new customer the list.  **Use the best practices we learned in this course.**  Show the **Java code** and the**required annotations** necessary to add a new customer to the list.    You may assume that a CustomerService class exists with the method addCustomer(Customer customer).   |  | | --- | |  | |
| Question 4 of 710.0 Points  **[15 minutes]**  Given is the following servlet code:  ***@WebServlet("/counter") public class CounterServlet extends HttpServlet {***  ***protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {         PrintWriter out = response.getWriter();***    ***// missing code***  ***out.println("<html>");         out.println("<body>");         out.println(value);         out.println("<form method = 'post' action = 'counter'>");         out.println("<input type = 'submit' name = 'result' value = 'increment'><br>");         out.println("</form>");          out.println("</body>");         out.println("</html>");***  ***out.flush();***  ***}***  ***}***  ***protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {         PrintWriter out = response.getWriter();***  ***// missing code***  ***out.println("<html>");         out.println("<body>");         out.println(value);         out.println("<form method = 'post' action = 'counter'>");         out.println("<input type = 'submit' name = 'result' value = 'increment'><br>");         out.println("</form>");         out.println("</body>");***  ***out.println("</html>");***  ***out.flush();***  ***}***  ***}***  The servlet shows a number that starts with 0, and every time you click the increment button, the number should increment. The shown number is unique for every user of the servlet.  Write the **missing code** so that the  output of the servlet should be as given below. **Only write the missing code!**   |  | | --- | |  |   //Get |
| Question 5 of 710.0 Points  **[10 minutes]**  Suppose you need to implement a client (front-end) application and a server (back-end) application. You want to use the REST protocol for the calls between the client and the server.  Someone advices you to use webflux instead of REST. The server (back-end) application uses a relational database and the driver we use to access the database is not reactive. Assume that there is no reactive driver for our database.  Even with a non-reactive database driver someone still advices you to use webflux instead of REST.  Explain clearly what advantage(s) you get when you use webflux instead of REST for the calls between the client and the server.   |  | | --- | |  | |
| Question 6 of 730.0 Points  **[30 minutes]**  Suppose we have already a working ProductService class with the methods:   * addProduct(Product product) * updateProduct(Product product)   A product has the attributes productNumber, name and price. Suppose we need to write a REST controller in front of this ProductService with 2 methods:   * addProduct(Product product) * updateProduct(Product product)   We also need to implement validation for our ProductController. The productNumber cannot be empty and the size should be between 5 and 8 characters. The name and price also cannot be empty.  a.    Write the code of the complete rest controller including all required annotations where **every individual method in the ProductController handles the validation errors**. Use the best practices we learned in this course.  b.    Write the code of the complete rest controller including all required annotations where the validation errors are handled in 1 place in the ProductController **for all methods of that specific controller**. Use the best practices we learned in this course and use the support that SpringMVC offers for this.  c.    Write the necessary code including all required annotations where the validation errors are handled in 1 place **for all methods of all controllers** that need to validate a Product. Use the best practices we learned in this course and use the support that SpringMVC offers for this.   |  | | --- | |  | |
| Question 7 of 75.0 Points  **[10 minutes]**  Describe how we can relate the concept of a**websocket**to the principles of Science & Technology of Consciousness (SCT). Your answer should be about **2 to 3 paragraphs (>10 lines of text)**, but should not exceed one page (handwritten). The number of points you get for this question depend on how well you explain the relationship between a**websocket**and the principles of SCT.   |  | | --- | |  |   WebSocket is asynchronous, non-blocking and bidirectional which is use in creating the real-time application such as Chat Application, Stock Trading System where data are required in real-time. The consciousness is a state of mind and body which can be achieved through Transcendental Meditation(TM) without any blocking and is real-time process. Websocket is bidirectional which means client and server exchange the update data with each-other. Similarly, SCT principles help to achieve the pure consciousness in the inner part of our body and mind. The regular follow of TM helps to feel the unified field of nature where all are equal. The human body interact with the nature to achieve the pure consciousness. |