/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Multiple Choice: 2 points each

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 1. What does REST stand for?

\* a. Really Easy State Transfer

\* b. Representational State Transfer

\* c. RAM Eclipse Standard Transfer

\* d. Read Easy State Trigger

\*/

char answer1 = ' b';

/\* 2. What does HTTP stand for?

\* a. HyperText Transfer Policy

\* b. Helios Typed Transfer Protocol

\* c. Happy Test Trasfer Policy

\* d. Hypertext Transfer Protocol

\*/

char answer2 = ' d';

/\* 3. Which HTTP method should be used to create a new resource?

\* a. PUT

\* b. POST

\* c. DELETE

\* d. GET

\*/

char answer3 = 'b ';

/\*

\* 4. Which HTTP method should be used to delete an existing resource?

\* a. DELETE

\* b. REMOVE

\* c. DISABLE

\* d. PATCH

\*/

char answer4 = ' a';

/\*

\* 5. Which HTTP method should be used to update an existing resource?

\* a. GET

\* b. RESET

\* c. UPDATE

\* d. PUT

\*/

char answer5 = ' d';

/\*

\* 6. Which HTTP method should be used to read an existing resource?

\* a. PATCH

\* b. RETRIEVE

\* c. READ

\* d. GET

\*/

char answer6 = ' d';

/\*

\* 7. When using Postman, what content-type have you been using to send POST requests to your APIs?

\* a. application/xml

\* b. text/plain

\* c. application/json

\* d. text/json

\*/

char answer7 = ' d';

/\*

\* 8. Who owns Maven?

\* a. Sun Microsystems

\* b. Apache

\* c. Oracle

\* d. Microsoft

\*/

char answer8 = 'b';

/\*

\* 9. What is Maven?

\* a. A build automation tool that also developers manage dependencies.

\* b. A scripting language.

\* c. An XML file where we list our project dependencies.

\* d. The tool used to compile and run SpringBoot applications.

\*/

char answer9 = ' d';

/\*

\* 10. In our project, where are Maven configurations set?

\* a. mvn.xml

\* b. pom.xml

\* c. application.properties

\* d. WEB-INF

\*/

char answer10 = ' b';

/\*

\* 11. What Spring annotation sets the base package to scan for components?

\* a. @ComponentScan

\* b. @Scanner

\* c. @SpringBootApplication

\* d. @Autowired

\*/

char answer11 = ' b';

/\*

\* 12. What Spring annotation assigns an endpoint mapping to method?

\* a. @Mapping

\* b. @Endpoint

\* c. @RequestMapping

\* d. @RestController

\*/

char answer12 = 'c ';

/\*

\* 13. What application layer contains our endpoint mappings?

\* a. Data Access

\* b. Repository

\* c. Service

\* d. Controller

\*/

char answer13 = ' b';

/\*

\* 14. What Spring annotation is used to inject dependencies?

\* a. @Autowired

\* b. @Inject

\* c. @Dependency

\* d. @AutoInject

\*/

char answer14 = 'c ';

/\*

\* 15. What Spring Data class can we extend to create a repository?

\* a. DAO

\* b. Repository

\* c. ServiceRepository

\* d. CrudRepository

\*/

char answer15 = ' b';

/\*

\* 16. Which of the following accurately portrays the proper flow of data through our API?

\* a. HTTP Request -> Controller -> Repository -> Database

\* b. HTTP Request -> Controller -> Service -> Repository -> Database

\* c. HTTP Request -> Database -> Repository -> Service -> Controller

\* d. HTTP Request -> Database -> Controller -> Service -> Repository

\*/

char answer16 = ' a';

/\*

\* 17. What application layer do I put business logic in?

\* a. Repository

\* b. Controller

\* c. Service

\* d. Server

\*/

char answer17 = 'a ';

/\*

\* 18. What does JPA stand for?

\* a. JavaScript Proprietary API

\* b. JSON Propagation API

\* c. Java Properties Application

\* d. Java Persistence API

\*/

char answer18 = ' d';

/\*

\* 19. What would you expect to happen when sending an HTTP POST request to this endpoint: `/customers`

\* a. An existing customer is updated.

\* b. A new customer is created.

\* c. A customer is retrieved from the database.

\* d. A customer is copied.

\*/

char answer19 = 'b ';

/\*

\* 20. Which of the following HTTP verb methods and endpoints would you expect to use to update an existing customer?

\* a. HTTP POST -> /customers/{id}

\* b. HTTP PUT -> /customers/{id}

\* c. HTTP PUT -> /customers

\* d. HTTP PUT -> /customers/update

\*/

char answer20 = 'b ';

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* true or false: 2 points each

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 21. API stands for Application Programming Interface

boolean answer21 = True;

// 22. JSON stands for JavaScript Object Notation.

boolean answer22 = True;

// 23. @Autowired is used to inject dependencies into a class.

boolean answer23 =True ;

// 24. With Spring Data, we can create our own repository by implementing the CrudRepository interface.

boolean answer24 = True;

// 25. Roy Fielding wrote the dissertation describing the REST architecture.

boolean answer25 =True ;

// 26. REST is a protocol to replace SOAP.

boolean answer26 = False;

// 27. REST is a specification for using HTTP to perform CRUD operations.

boolean answer27 = False;

// 28. `@ComponentScan("com.promineotech.controller")` will find components in the package `com.promineotech.controller.sub`.

boolean answer28 = True;

// 29. We need to put the @RestController annotation in every class we create in a SpringBoot application.

boolean answer29 = False;

// 30. The findAll method on CrudRepository will return an Iterable of whatever type was specified when creating the repository.

boolean answer30 =True ;

// 31. The update method on CrudRepository is used to update an existing entity.

boolean answer31 =True ;

// 32. You can save an entity by calling the save method on the instance of the entity you wish to save.

boolean answer32 = False;

// 33. `@RequestMapping("/products/{id}", method=RequestMethod.PUT)` is a valid use of the @RequestMapping annotation.

boolean answer33 = False;

// 34. When we are connecting our API to a database, the fields on our entity classes should use the snake case naming convention (i.e. first\_name)

boolean answer34 = False;

// 35. An HTTP POST request can only create something new, it could never be used to delete something.

boolean answer35 = True;

// 36. Developers decide what HTTP requests and endpoints to use for any operation, however, they should follow standards.

boolean answer36 =True ;

// 37. APIs can only accept JSON in a request's payload/body.

boolean answer37 = True;

// 38. Maven installs dependencies for our project based on the dependencies listed in application.properties.

boolean answer38 = True;

// 39. Putting business logic in the service layer of an application increases code reusibility.

boolean answer39 = True;

// 40. If you wanted to build an endpoint that accepted an HTTP GET request to delete data, you could.

boolean answer40 =True (can but frowned on) ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Essay Questions: 5 points each

\*

\* Use the employees database schema to answer these questions.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 41. Write an endpoint method that is mapped to the endpoint `/customer`. This endpoint should be used to save a new

\* Customer. In the body of the method put only the line `return service.save(customer);`.

\*

\*/

Post= /Customer/ new customer/ {id}

This will create a new user and save it by the id it is assigned.

Public Customer createCustomer ( Customer customer){

Return service.save(customer);

}

/\* 42. Write an endpoint method that is mapped to the endpoint `/customer/{id}`. This endpoint should be used to retrieve a

\* specific Customer. In the body of the method, put only the line `return service.find(id);`.

\*

\*/ public Customer getCustomer(Long id) {

Return service.fin(id);

}

/\* 43. Write an endpoint method that is mapped to the endpoint `/customer/{id}`. This endpoint should be used to update a

\* specific Customer. In the body of the method, put only the line `return service.update(id, customer);`.

\*

\*/ public Customer update Customer (Long id){

Return service.update(id, customer);

}

/\* 44. Write an interface that extends CrudRepository<Product, Long> and would function as repository for a class called Product.

\*

\*/