

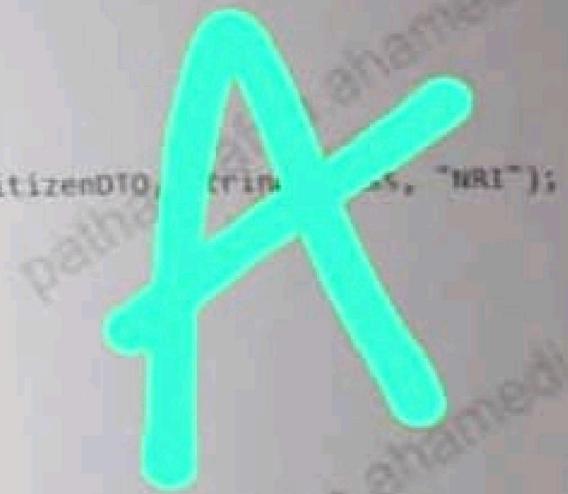
## OBJECTIVE QUESTIONS

1. All the necessary imports and configurations are done.

Choose the correct url that can be placed at Line 1 in the below code snippet to get the ou

```
RestTemplate restTemplate = new RestTemplate();
CitizenDTO citizenDTO = new CitizenDTO();
citizenDTO.setCitizenId("1201");
citizenDTO.setName("Nick Stinson");
citizenDTO.setAge(32);
citizenDTO.setPassportNumber("A123456");
//Line 1
String response = restTemplate.postForObject(url, citizenDTO, String.class, "NRI");
System.out.println(response);
```

- String url="http://localhost:5555/citizens/citizenType";
- String url="http://localhost:5555/{citizenType}";
- String url="http://localhost:5555/citizens/{citizenType}";
- String url="/citizens/{citizenType}";



### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
list.add(c1);
list.add(c2);
list.add(c3);
list.add(c4);
if (list.contains(c3)) {
    list.set(1, list.get(3));
}
Stream<Trainee> stream = list.stream().filter(trainee -> trainee.getTraineeName().length() > 10);
stream.forEach((trainee) -> System.out.print(trainee.getTraineeId() + " "));
```

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115

Reset

Save

A D

### Question 1

Which of the following method is properly mapped to handle the HTTP GET request with URL `/students/S1101` ?

Choose TWO correct options.

- `@RequestMapping("/students/{studentId}",method=RequestMethod.GET)  
public void getDetails(@PathVariable String studentId){}`
- `@RequestMapping("/students")  
public void getDetails(@PathVariable String studentId){}`
- `@RequestMapping("/students")  
public void getDetails(@RequestParam String studentId){}`
- `@GetMapping("/students/{studentId}")  
public void getDetails(@PathVariable String studentId){}`

Reset

Save

```
public static void main(String[] args) {  
    SpringApplication.run(SpringBootApplication.class, args);  
}  
  
public void run(String... args) throws Exception {  
    System.out.println(employeeService.performance(19));  
}  
}  
  
○ service.INCORRECT_BONUS  
○ Your performance was satisfactory  
○ Your performance was satisfactory  
○ Your performance was excellent  
○ Your performance was satisfactory
```

Reset

Save

```
@GeneratedValue(generator="pkgen", strategy=GenerationType.TABLE)
private Integer patientId;
private String patientName;
private long patientPhoneNumber;
//getters and setters
```

Assumption:

1. All necessary imports and configurations are done

What will be the value of pat\_id in id\_gen table after persisting details of three patients in PATIENT table?

- 110
- 117
- 103
- 115

Reset

Save

D

```
@Override  
public void run(String... args) throws Exception {  
    service.callingRepositoryClass();  
}  
}
```

- PharmacyService constructor called  
PharmacyRepository constructor called
- PharmacyService constructor called
- PharmacyRepository constructor called  
PharmacyRepository constructor called
- PharmacyRepository constructor called
- PharmacyRepository constructor called
- PharmacyService constructor called  
PharmacyRepository constructor called

Reset

Save

Assumptions:

1. All necessary imports and configurations are done
2. Application is running on port 8080
3. LoanDTO is a Java bean class

What will be the output when the following code is executed?

```
LoanDTO loanDTO = new LoanDTO();
loanDTO.setLoanId(1001);
String url = "http://localhost:8080/houseLoan/{loanId}";
RestTemplate restTemplate = new RestTemplate();
restTemplate.put(url, loanDTO, loanDTO.getLoanId());
```

- Loan deleted successfully for : 1001
- null
- IllegalStateException will be thrown
- Loan deleted successfully for : null

### Question 3

What will be the output of the code given below? Note: Consider all necessary imports are added.

```
public static void main(String[] args) {  
    LocalTime time = LocalTime.of(22, 58, 15);  
    time.plusMinutes(5);  
    time = time.plusHours(1);  
    System.out.println("Time - " + time);  
}
```

- Time-22:58:15
- Time-00:03:15
- Time-23:03:15
- Time-23:58:15

Reset

Save

DONE



CUSTOMERRepository.delete(customer);

- It will delete customer with customer\_id 1 from CUSTOMER table but no instruments from the INSTRUMENT table.
- It will throw SQLIntegrityConstraintViolationException exception.
- It will delete customer with customer\_id 1 from CUSTOMER table and instrument with instrument\_id 1 from INSTRUMENT table.
- It will delete customer with customer\_id 1 from CUSTOMER table and instruments with instrument\_id 1 from INSTRUMENT table.

Reset

Save

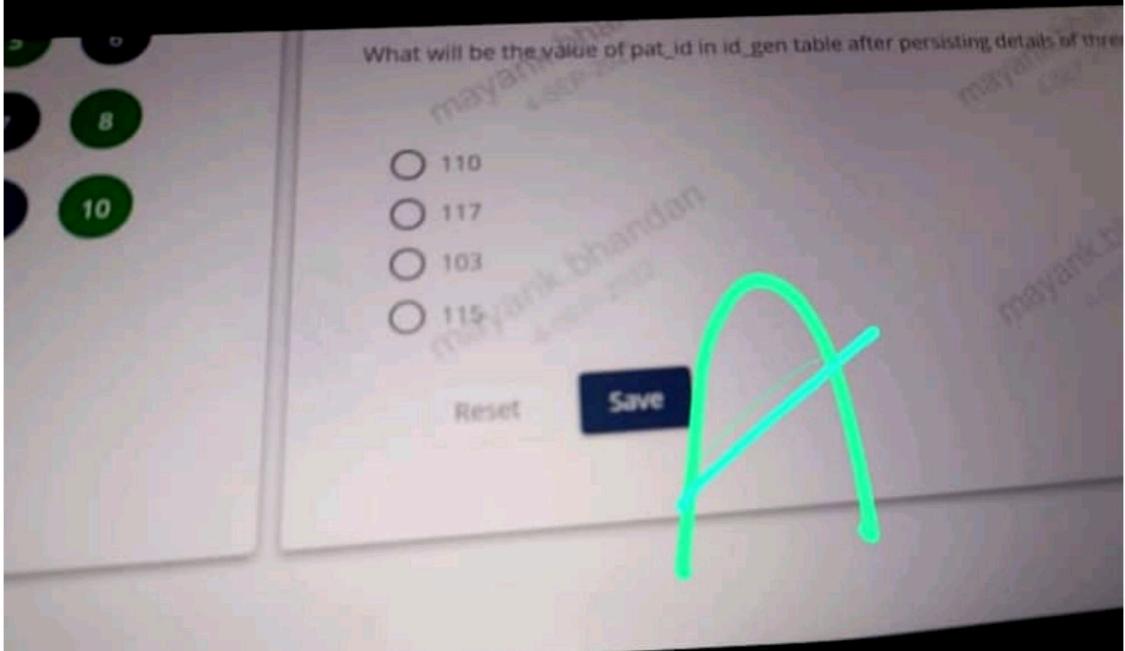
A



X



T



Swipe up for filters

Add a caption...

(1)



DONE



02 : 50 : 49  
Hours Minutes Seconds

You're sharing Q... Chrome

OBJECTIVE QUESTIONS

Focus Area 1



```
public class Validator{  
    public static boolean validatePassword(String password){  
        if(password.length() < 6 || !password.matches("[a-zA-Z]+"))  
            return false;  
        return true;  
    }  
}
```

Following is the test case written for testing the above Validator class. Identify:

```
public class ValidatorTest{  
    @Test  
    public void validatePasswordTest(){  
        //  
    }  
}
```

Which of the following assertion statements would be appropriate at line 1 for valid test cases. Choose TWO correct options.

- assertEquals(true, validate("ABC123"));
- assertEquals(true, validate("Abc123"));
- assertEquals(true, validate("Abc123!"));
- assertEquals(true, validate("Abc123@"));

You're being proctored!



02 : 50 : 47

Hours Minutes Seconds

You're sharing Google Chrome

OBJECTIVE QUESTIONS

Focus Area 1



```
public static void main(String[] args) {  
    Map<String, String> nameMap = new HashMap<String, String>();  
  
    nameMap.put("A", "Andre");  
    nameMap.put("B", "Bob");  
    nameMap.put(null, null);  
    nameMap.put("C", "Catlyn");  
    nameMap.put(new String("A"), "Avan");  
    System.out.println(nameMap);  
}
```

Assumption:

1. All necessary configuration and imports are done



- (null=null, A=Avan, B=Bob, C=Catlyn)
- (A=Avan, B=Bob, C=Catlyn)
- The code will throw run time exception as the same key cannot be used for 2 values.
- The code will throw NullPointerException as the HashMap cannot have null as key value.

Reset

Save

DONE



customerRepository.delete(customer);

It will delete customer with customer\_id 1 from CUSTOMER table but no instruments from INSTRUMENT table.

It will throw SQLIntegrityConstraintViolationException exception.

It will delete customer with customer\_id 1 from CUSTOMER table and instrument with instrument\_id 1 from INSTRUMENT table.

It will delete customer with customer\_id 1 from CUSTOMER table and instruments with instrument\_id 1 from INSTRUMENT table.

Reset      Save

C

## Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
RestTemplate restTemplate = new RestTemplate();
CitizenDTO citizenDTO = new CitizenDTO();
citizenDTO.setCitizenId("1201");
citizenDTO.setName("Nick Stinson");
citizenDTO.setAge(32);
citizenDTO.setPassportNumber("A123456");
//Line 1
String response = restTemplate.postForObject(url, citizenDTO, String.class, "NRI");
System.out.println(response);
```

- String url="http://localhost:5555/citizens/citizenType";
- String url="http://localhost:5555/{citizenType}";
- String url="http://localhost:5555/citizens/{citizenType}";
- String url="/citizens/{citizenType}";

Reset

Save

### Question 8

Which of the following method is properly mapped to handle the HTTP GET request with URI /students/S1101 ?

Choose TWO correct options.

- `@RequestMapping("/students")  
public void getDetails(@RequestParam String studentId){}`
- `@RequestMapping("/students")  
public void getDetails(@PathVariable String studentId){}`
- `@RequestMapping("/students/{studentId}",method=RequestMethod.GET)  
public void getDetails(@PathVariable String studentId){}`
- `@GetMapping("/students/{studentId}")  
public void getDetails(@PathVariable String studentId){}`

Reset

Save



Consider the below code.

```
public class CollectionsTest {  
    public static void main(String[] args) {  
        List<String> numList = new ArrayList<>();  
        int listSize = numList.size();  
        listSize += 10;  
        numList.add("Java");  
        for (int i = 1; i < listSize; i++) {  
            numList.add(numList.get(i - 1) + "-Java");  
            numList.add(++i, "Java");  
        }  
        numList.remove(2);  
        System.out.println("Element at 3rd position: " + numList.get(2));  
        System.out.println("Size: " + numList.size());  
    }  
}
```

What will be the output of above code snippet? Note: Consider all necessary imports are added.

- Element at 3rd position: Java-Java  
Size: 9
- Element at 3rd position: Java  
Size: 9

size

Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

PRODUCT (product\_id is the primary key)

product_id	product_name
2001	Laptop
2002	Refrigerator

What will be the value of productId generated, when a new row is inserted in PRODUCT table?

- 2003
- 0001
- 1
- 2000

Reset

Save

Which of the following options are true with respect to HTML form elements? (Select any Two)

- minlength='4' means the user can enter only value '4'
- id attribute must be kept unique for every elements inside form tag
- Input field with 'type' as 'number' also allows alphabets
- Name attribute of radio buttons should be same, to make only one radio button of them to be selectable

Reset

Save

B . D

Focus Area 1

- 1
- 2
- 3
- 4
- 5

Question 3

What will be the output of the code given below? Note: Consider all necessary imports are added.

```
public static void main(String[] args) {  
    LocalTime time = LocalTime.of(22, 58, 15);  
    time.plusMinutes(5);  
    time=time.plusHours(1);  
    System.out.println("Time-"+time);  
}
```

38% 62%

- Time-22:58:15
- Time-23:58:15
- Time-00:03:15
- Time-23:03:15

Reset

Save

B

You're be

## OBJECTIVE QUESTIONS

1. All the necessary imports and configurations are done.

Choose the correct url that can be placed at Line 1 in the below code snippet to get the ou

```
RestTemplate restTemplate = new RestTemplate();
CitizenDTO citizenDTO = new CitizenDTO();
citizenDTO.setCitizenId("1201");
citizenDTO.setName("Nick Stinson");
citizenDTO.setAge(32);
citizenDTO.setPassportNumber("A123456");
//Line 1
String response = restTemplate.postForObject(url, citizenDTO, String.class, "NRI");
System.out.println(response);
```

- String url="http://localhost:5555/citizens/citizenType";
- String url="http://localhost:5555/{citizenType}";
- String url="http://localhost:5555/citizens/{citizenType}";
- String url="/citizens/{citizenType}";

Focus Area 1

- 1
- 2
- 3
- 4
- 5

Which of the following testcases to test validity of customer name will pass? (Choose two correct options)

- ```
@Test  
public void validateCustomerNameValidTest1() {  
    Assertions.assertTrue(Validator.isValidCustomerName("Aabel james"));  
}
```
- ```
@Test  
public void validateCustomerNameValidTest2() {  
    Assertions.assertEquals(Validator.isValidCustomerName("Aabel ja"), true);  
}
```
- ```
@Test  
public void validateCustomerNameValidTest3() {  
    Assertions.assertFalse(Validator.isValidCustomerName("Aabel ja."));  
}
```
- ```
@Test  
public void validateCustomerNameValidTest4() {  
    Assertions.assertEquals(Validator.isValidCustomerName("Aabel james"), false);  
}
```



## OBJECTIVE QUESTIONS

### Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
public void run(String... args) throws Exception {  
    System.out.println(environment.getProperty("welcome")); //Line1  
    System.out.println(environment.getProperty("hello"));  
}
```

- Welcome to Spring Boot
- Hello World
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- I am a Spring Boot Application
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- Exception generated at Line 1 due to ambiguity in the key "welcome"

Reset

Save

### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
list.add(c1);
list.add(c2);
list.add(c3);
list.add(c4);
if (list.contains(c3)) {
    list.set(1, list.get(3));
}
Stream<Trainee> stream = list.stream().filter(trainee -> trainee.getTraineeName().length() > 10);
stream.forEach((trainee) -> System.out.print(trainee.getTraineeId() + " "));
```

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115

Reset

Save

### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
list.add(c1);
list.add(c2);
list.add(c3);
list.add(c4);
if (list.contains(c3)) {
    list.set(1, list.get(3));
}
stream<Trainee> stream = list.stream().filter(trainee -> trainee.getTraineeName().length()
stream.forEach((trainee) -> System.out.print(trainee.getTraineeId() + " "));
```

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115

Reset

Save

Assumptions:

1. All the necessary imports and configurations are done
2. Port number used is 8765

What will be the output when a HTTP POST request is made to the above controller with url as <http://localhost:8765/infy> and below data?

```
{  
    "employeeId": 1,  
    "emailId": "andrew@infy.com",  
    "employeeName": "Andrew Wyatt",  
    "designation": "Senior Associate"  
}
```

- EmployeeDTO [employeeId=null, emailId=null, name=null, designation=null]
- EmployeeDTO [employeeId=1, emailId=andrew@infy.com, name=null, designation=Senior Associate]
- EmployeeDTO [employeeId=1, emailId= andrew@infy.com, name=Andrew Wyatt, designation= Senior Associate]
- 404 NOT\_FOUND error

Reset

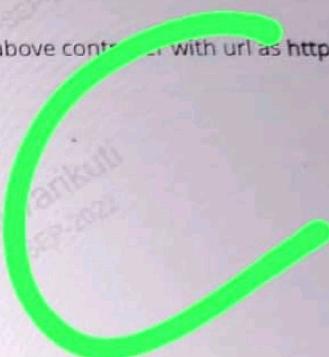
Save

Assumptions:

1. All the necessary imports and configurations are done
2. Port number used is 8765

What will be the output when a HTTP POST request is made to the above controller with url as <http://localhost:8765/infy> and below data?

```
{  
    "employeeId": 1,  
    "emailId": "andrew@infy.com",  
    "employeeName": "Andrew Wyatt",  
    "designation": "Senior Associate"  
}
```



- EmployeeDTO [employeeId=null, emailId=null, name=null, designation=null]
- EmployeeDTO [employeeId=1, emailId=andrew@infy.com, name=null, designation=Senior Associate]
- EmployeeDTO [employeeId=1, emailId= andrew@infy.com, name=Andrew Wyatt, designation= Senior Associate]
- 404 NOT\_FOUND error

Reset

Save

```
trainee_id int auto_increment primary key,  
name varchar(30),  
phone_number varchar(10),  
date_of_joining date  
);
```

Assumption:

1. All the necessary imports and configurations are done.

Which of the following statements can be inserted at **Line 1** to generate primary key values for traineeId?

- @GeneratedValue(strategy = GenerationType.IDENTITY)
- @GeneratedValue(strategy = GenerationType.AUTO)
- @GeneratedValue(strategy = GenerationType.SEQUENCE)
- @GeneratedValue(strategy = GenerationType.TABLE)

Reset

Save

## Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
public void run(String... args) throws Exception {  
    System.out.println(environment.getProperty("welcome")); //Line1  
    System.out.println(environment.getProperty("hello"));  
}
```

- Welcome to Spring Boot
- Hello World
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- I am a Spring Boot Application
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- Exception generated at Line 1 due to ambiguity in the key "welcome"

Reset

Save

What will be the output of the code given below? Note: Consider all necessary imports are added.

```
public static void main(String[] args) {  
    LocalTime time = LocalTime.of(22, 58, 15);  
    time.plusMinutes(5);  
    time=time.plusHours(1);  
    System.out.println("Time-"+time);  
}
```

- Time-00:03:15
- Time-23:58:15
- Time-22:58:15
- Time-23:03:15

Reset

Save

```
} naga  
} stream trainee> stream = list.stream()  
stream.forEach((trainee) -> System.out.  
}  
}
```

nagaraju  
5-SEP-2022

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115



Reset

Save

02 : 55 : 23  
Hours Minutes Seconds

### OBJECTIVE QUESTIONS

1. All the necessary imports and configurations are done.

Choose the correct url that can be placed at Line 1 in the below code snippet to get the ou

```
RestTemplate restTemplate = new RestTemplate();
CitizenDTO citizenDTO = new CitizenDTO();
citizenDTO.set公民ID("1201");
citizenDTO.setName("Nick Stinson");
citizenDTO.setAge(32);
citizenDTO.setPassportNumber("A123456");
//Line 1
String response = restTemplate.postForObject(url, citizenDTO, String.class, "MRI");
System.out.println(response);
```

- String url="http://localhost:5555/citizens/citizenType";
- String url="http://localhost:5555/{citizenType}";
- String url="http://localhost:5555/citizens/{citizenType}";
- String url="/citizens/{citizenType}";

## OBJECTIVE QUESTIONS

UKulele	Peru
Violin	Havana

Assumptions:

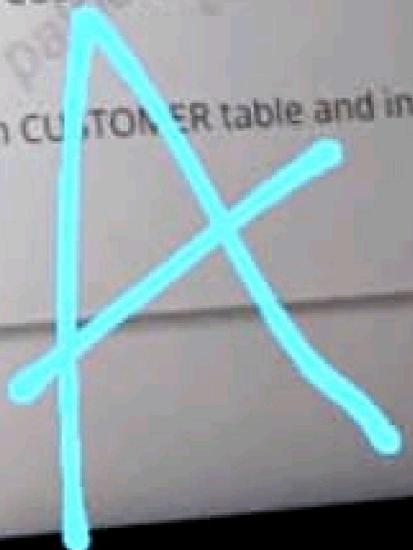
1. All necessary imports and configurations are done.
2. customerRepository is a valid reference of CustomerRepository which extends CrudRepository.
3. Transaction is managed by Spring.

What will be the output when the following code is executed?

```
Customer customer = customerRepository.findById(1).get();
customerRepository.delete(customer);
```



- It will delete customer with customer\_id 1 from CUSTOMER table but no instrument.
- It will throw SQLIntegrityConstraintViolationException exception.
- It will delete customer with customer\_id 1 from CUSTOMER table and instrument INSTRUMENT table.
- It will delete customer with customer\_id 1 from CUSTOMER table and instrument Guitar from INSTRUMENT table.



```
vivek.sharma68 4-SEP-2022
for (String element : set) {
    System.out.print(element+" ");
}
}
```

Assumption:

1. All necessary configuration and imports are done

Note: The elements might appear in un-ordered fashion

- Earth Mars Jupiter Saturn
- Earth Mars Saturn
- Saturn Jupiter Mars Earth
- Saturn Jupiter Earth

Reset

Save

What will be the output of the below code snippet?

```
public class DateTester {  
    public static void main(String[] args) {  
        DayOfWeek dayOfWeek = DayOfWeek.of(7);  
        System.out.println(dayOfWeek);  
    }  
}
```

Assumption:

1. All necessary imports are done

- SATURDAY
- SUNDAY
- DateTimeException exception is thrown
- IndexOutOfBoundsException exception is thrown

Reset

Save

What will be the output of the code given below? Note: Consider all necessary imports are added.

```
public static void main(String[] args) {  
    LocalTime time = LocalTime.of(22, 58, 15);  
    time.plusMinutes(5);  
    time=time.plusHours(1);  
    System.out.println("Time-"+time);  
}
```

- Time-00:03:15
- Time-23:58:15
- Time-22:58:15
- Time-23:03:15

Reset

Save

Product	Category
2001	Laptop
2002	Refrigerator

What will be the value of productId generated, when a new row is inserted in PRODUCT table?

- 2003
- 0001
- 1
- 2000

Reset

Save

2003

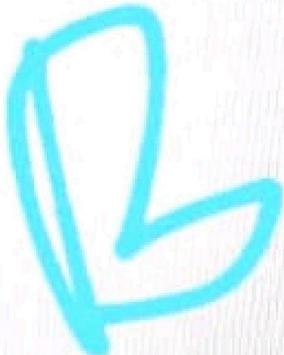
```
System.out.println(hour+" "+sec)
```

```
}
```

### Assumption:

1. All necessary configuration and imports are

- 9:45:59
- 9:59:45
- 09:45:59
- 09:59:45



Reset

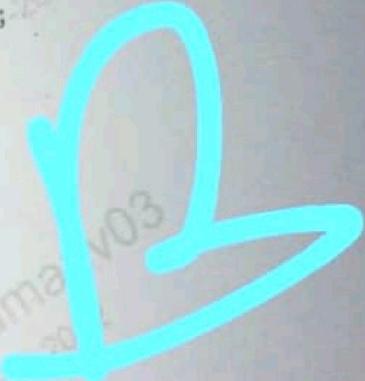
Save

2. 8765 is the port number used

What should be inserted at line 1 in below code to delete the record of a student on the basis of studentId?

```
String url = "http://localhost:8765/students/{studentId}";  
RestTemplate restTemplate = new RestTemplate();  
//line 1
```

- restTemplate.deleteForObject(url, StudentDTO.class, studentId); 
- restTemplate.delete(url, StudentDTO.class, studentId);
- restTemplate.delete(url, studentId);
- restTemplate.deleteForObject(url, studentId);



Reset

Save

What will be the output of the code given below? Note: Consider all necessary imports are added.

```
public static void main(String[] args) {  
    LocalTime time = LocalTime.of(22, 58, 15);  
    time.plusMinutes(5);  
    time=time.plusHours(1);  
    System.out.println("Time-"+time);  
}
```

- Time-00:03:15
- Time-23:58:15
- Time-22:58:15
- Time-23:03:15

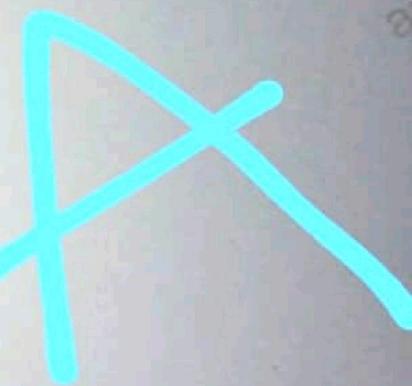
Reset

Save

```
// Line1  
public class ExceptionHandler {  
    // Line2  
    public ResponseEntity<ErrorInfo> TravelBookingExceptionHandler(TravelBookingException exce  
        //rest of the code  
    }  
}
```

Which annotations can be inserted at Line1 and Line2 so that ExceptionHandler class can handle the appropriate application? Note: Consider all necessary imports are added.

- Line1 - @RestController  
Line2 - @ExceptionHandler(TravelBookingException.class)
- Line1 - @RestController  
Line2 - @ExceptionHandler()
- Line1 - @RestControllerAdvice  
Line2 - @ExceptionHandler(TravelBookingException.java)
- Line1 - @RestControllerAdvice  
Line2 - @ExceptionHandler(TravelBookingException.class)



```
}

@Override
public void run(String... args) throws Exception {
    System.out.println(environment.getProperty("welcome")); //Line1
    System.out.println(environment.getProperty("hello"));
}

}

}

```

- Welcome to Spring Boot
- Hello World
  - I am a Spring Boot Application
  - Welcome to DemoApplication
  - Hello World
  - I am a Spring Boot Application
  - I am a Spring Boot Application
  - Welcome to DemoApplication
  - Hello World
  - Exception generated at Line 1 due to ambiguity in the key "welcome"

### Focus Area 1



```
public static void main(String[] args) {  
    Map<String, String> nameMap = new HashMap<String, String>();  
  
    nameMap.put("A", "Andre");  
    nameMap.put("B", "Bob");  
    nameMap.put(null, null);  
    nameMap.put("C", "Catlyn");  
    nameMap.put(new String("A"), "Avan");  
    System.out.println(nameMap);  
}
```

Assumption:

1. All necessary configuration and imports are done

- (null=null, A=Avan, B=Bob, C=Catlyn)
- (A=Avan, B=Bob, C=Catlyn)
- The code will throw run time exception as the same key cannot be used for 2 values.
- The code will throw NullPointerException as the HashMap cannot have null as key value.

Reset

Save

Consider the below code.

```
public class CollectionsTest {  
    public static void main(String[] args) {  
        List<String> numList = new ArrayList<>();  
        int listSize = numList.size();  
        listSize += 10;  
        numList.add("Java");  
        for (int i = 1; i < listSize; i++) {  
            numList.add(numList.get(i - 1) + "-Java");  
            numList.add(++i, "Java");  
        }  
        numList.remove(2);  
        System.out.println("Element at 3rd position: " + numList.get(2));  
        System.out.println("Size: " + numList.size());  
    }  
}
```

What will be the output of above code snippet? Note: Consider all necessary imports are added.

- Element at 3rd position: Java-Java  
Size: 9
- Element at 3rd position: Java  
Size: 9



Consider the following routes:

```
const routes : Routes = [
  { path: '', redirectTo: '/home' , pathMatch: 'full' },
  { path: 'home', component: HomeComponent },
  { path : 'search/:name', component: SearchComponent }
];
```

Assume application is hosted on localhost:4200, which of the below path combinations will work?

- i) http://localhost:4200/
- ii) http://localhost:4200/home
- iii) http://localhost:4200/search/Alex

- only 2 and 3 will work
- All URL's 1,2 and 3 will work
- only 1 and 2 will work
- only 1 will work

Reset

Save

Scanned with CamScanner

## Front End Basics

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15

What is the default list style for an unordered list <ul> ?

- rectangle
- circle
- disc
- square

Reset

Save



Note: The bottom boxes are grids for understanding purpose and are not part of required output  
 Which of the following code snippets will render the desired output ?

- < div class="col-md-4 offset-md-1" style="background-color: yellow" >BOX 1</div >
- < div class="col-md-3" style="background-color: skyblue" >BOX 2</div >  
 < div class="col-md-4 offset-md-2" style="background-color: coral" >BOX 3</div >
- < div class="col-md-4 offset-md-1" style="background-color: yellow" >BOX 1</div >
- < div class="col-md-4" style="background-color: skyblue" >BOX 2</div >  
 < div class="col-md-4 offset-md-1" style="background-color: coral" >BOX 3</div >
- < div class="col-md-5 offset-md-1" style="background-color: yellow" >BOX 1</div >
- < div class="col-md-4" style="background-color: skyblue" >BOX 2</div >  
 < div class="col-md-6 offset-md-6" style="background-color: coral" >BOX 3</div >
- < div class="col-md-4 offset-md-2" style="background-color: yellow" >BOX 1</div >
- < div class="col-md-3" style="background-color: skyblue" >BOX 2</div >  
 < div class="col-md-6 offset-md-2" style="background-color: coral" >BOX 3</div >

```
RestTemplate restTemplate = new RestTemplate();
CitizenDTO citizenDTO = new CitizenDTO();
citizenDTO.setCitizenId("1201");
citizenDTO.setName("Nick Stinson");
citizenDTO.setAge(32);
citizenDTO.setPassportNumber("A123456");
//Line 1
String response = restTemplate.postForObject(url, citizenDTO, String.class, "NRI");
System.out.println(response);
```

- String url="http://localhost:5555/citizens/citizenType";
- String url="http://localhost:5555/{citizenType}";
- String url="http://localhost:5555/citizens/{citizenType}";
- String url="/citizens/{citizenType}";

Reset

Save

## certification - Notepad

File Edit Format View Help

1.Assertions.assertTrue(Validator.validatePassword("Ab1"));  
Assertions.assertEquals(Boolean.TRUE,Validator.validatePassword("Abc123"));

2.Element at 3rd position :java-java  
Size:10

3. 110 112 (trainee class)

4.SUNDAY (DateTester)

5.NullPointerException(MapTester)

6.option 1(Bean validation API)

7.2003(product Id)

8.Option 1,2,4 (Manufactured Date)

I

9.String url ="http://localhost:5555/citizens/{citizenType}"

10.Time-23:58:15

11.InvalidAmount() test case:fails,ValidAmount() test case: fails

12.Earth  
Mars  
Jupiter  
Saturn

13.{null=null, A=Avan, B=Bob, C=Catlyn}

14.option1(date of joining)--doubt

15.9:59:45

16.option1 (StudentId,name,dob - bean validation)

17.Identity strategy (Trainee)

18.



Type here to search



Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

//Getters and setters

PRODUCT (product\_id is the primary key)

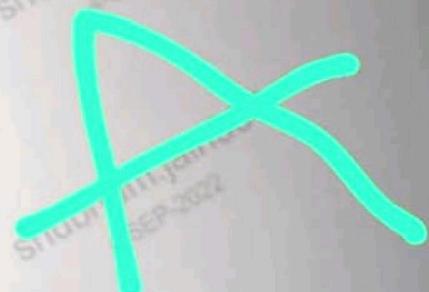
product_id	product_name
2001	Laptop
2002	Refrigerator

What will be the value of productId generated, when a new row is inserted in PRODUCT table?

- 2003
- 0001
- 1
- 2000

Reset

Save



Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
list.add(c1);
list.add(c2);
list.add(c3);
list.add(c4);
if (list.contains(c3)) {
    list.set(1, list.get(3));
}
stream<Trainee> stream = list.stream().filter(trainee -> trainee.getTraineeName().length()
stream.forEach((trainee) -> System.out.print(trainee.getTraineeId() + " "));
```

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115

Reset

Save

Assumptions:

1. All the necessary imports and configurations are done
2. Port number used is 8765

What will be the output when a HTTP POST request is made to the above controller with url as <http://localhost:8765/infy> and below data?

```
{  
    "employeeId": 1,  
    "emailId": "andrew@infy.com",  
    "employeeName": "Andrew Wyatt",  
    "designation": "Senior Associate"  
}
```

- EmployeeDTO [employeeId=null, emailId=null, name=null, designation=null]
- EmployeeDTO [employeeId=1, emailId=andrew@infy.com, name=null, designation=Senior Associate]
- EmployeeDTO [employeeId=1, emailId= andrew@infy.com, name=Andrew Wyatt, designation= Senior Associate]
- 404 NOT\_FOUND error

Reset

Save

## Focus Area 2

- |   |    |
|---|----|
| 1 | 2  |
| 3 | 4  |
| 5 | 6  |
| 7 | 8  |
| 9 | 10 |

```
public void run(String... args) throws Exception {  
    System.out.println(environment.getProperty("welcome")); //Line1  
    System.out.println(environment.getProperty("hello"));  
}
```

- Welcome to Spring Boot
- Hello World
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- I am a Spring Boot Application
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- Exception generated at Line 1 due to ambiguity in the key "welcome"

Reset

Save

Focus Area 1

- 1
- 2
- 3
- 4
- 5

Which of the following testcases to test validity of customer name will pass? (Choose two correct options)

```
@Test  
public void validateCustomerNameValidTest1() {  
    Assertions.assertTrue(validator.isValidCustomerName("Aabel james"));  
}  
  
@Test  
public void validateCustomerNameValidTest2() {  
    Assertions.assertEquals(validator.isValidCustomerName("Aabel james"),true);  
}  
  
@Test  
public void validateCustomerNameValidTest3() {  
    Assertions.assertFalse(validator.isValidCustomerName("Aabel james"));  
}  
  
@Test  
public void validateCustomerNameValidTest4() {  
    Assertions.assertEquals(validator.isValidCustomerName("Aabel james"),false);  
}
```

Focus Area 1

- 1
- 2
- 3
- 4
- 5

Question 3

What will be the output of the code given below? Note: Consider all necessary imports are added.

```
public static void main(String[] args) {  
    LocalTime time = LocalTime.of(22, 58, 15);  
    time.plusMinutes(5);  
    time=time.plusHours(1);  
    System.out.println("Time-"+time);  
}
```

- Time-22:58:15
- Time-23:58:15
- Time-00:03:15
- Time-23:03:15

Reset

Save

38% 62%

You're be

## Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
Create Table trainee (
    trainee_id int auto_increment primary key,
    name varchar(30),
    phone_number varchar(10),
    date_of_joining date
);
```

Assumption:

1. All the necessary imports and configurations are done.

Which of the following statements can be inserted at **Line 1** to generate primary key values for trainee?

- @GeneratedValue(strategy = GenerationType.IDENTITY)
- @GeneratedValue(strategy = GenerationType.AUTO)
- @GeneratedValue(strategy = GenerationType.SEQUENCE)
- @GeneratedValue(strategy = GenerationType.TABLE)

Reset

Save



## OBJECTIVE QUESTIONS

### Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
public void run(String... args) throws Exception {  
    System.out.println(environment.getProperty("welcome")); //Line1  
    System.out.println(environment.getProperty("hello"));  
}
```

- Welcome to Spring Boot
- Hello World
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- I am a Spring Boot Application
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- Exception generated at Line 1 due to ambiguity in the key "welcome"

Reset

Save

### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
list.add(c1);
list.add(c2);
list.add(c3);
list.add(c4);
if (list.contains(c3)) {
    list.set(1, list.get(3));
}
stream<Trainee> stream = list.stream().filter(trainee -> trainee.getTraineeName().length()
stream.forEach((trainee) -> System.out.print(trainee.getTraineeId() + " "));
```

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115

Reset

Save

## Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
Create Table trainee (
    trainee_id int auto_increment primary key,
    name varchar(30),
    phone_number varchar(10),
    date_of_joining date
);
```

Assumption:

1. All the necessary imports and configurations are done.

Which of the following statements can be inserted at **Line 1** to generate primary key values for trainee?

- @GeneratedValue(strategy = GenerationType.IDENTITY)
- @GeneratedValue(strategy = GenerationType.AUTO)
- @GeneratedValue(strategy = GenerationType.SEQUENCE)
- @GeneratedValue(strategy = GenerationType.TABLE)

Reset

Save

Focus Area 2

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

```
create table trainee (
    trainee_id int auto_increment primary key,
    name varchar(30),
    phone_number varchar(10),
    date_of_joining date
);
```

Assumption:

1. All the necessary imports and configurations are done.

Which of the following statements can be inserted at Line 1 to generate primary key values for traineeId?

- @GeneratedValue(strategy = GenerationType.IDENTITY)
- @GeneratedValue(strategy = GenerationType.AUTO)
- @GeneratedValue(strategy = GenerationType.SEQUENCE)
- @GeneratedValue(strategy = GenerationType.TABLE)

Reset

Save

Focus Area 2

- |   |    |
|---|----|
| 1 | 2  |
| 3 | 4  |
| 5 | 6  |
| 7 | 8  |
| 9 | 10 |

```
public void run(String... args) throws Exception {  
    System.out.println(environment.getProperty("welcome")); //Line1  
    System.out.println(environment.getProperty("hello"));  
}
```

- Welcome to Spring Boot
- Hello World
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- I am a Spring Boot Application
- I am a Spring Boot Application
- Welcome to DemoApplication
- Hello World
- Exception generated at Line 1 due to ambiguity in the key "welcome"

Reset

Save

Assumptions:

1. All the necessary imports and configurations are done
2. Port number used is 8765

What will be the output when a HTTP POST request is made to the above controller with url as http://localhost:8765/infy and below data?

```
{  
    "employeeId": 1,  
    "emailId": "andrew@infy.com",  
    "employeeName": "Andrew Wyatt",  
    "designation": "Senior Associate"  
}
```

- EmployeeDTO [employeeId=null, emailId=null, name=null, designation=null]
- EmployeeDTO [employeeId=1, emailId=andrew@infy.com, name=null, designation=Senior Associate]
- EmployeeDTO [employeeId=1, emailId= andrew@infy.com, name=Andrew Wyatt, designation= Senior Associate]
- 404 NOT\_FOUND error

Reset

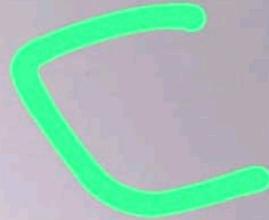
Save

Assumptions:

1. All the necessary imports and configurations are done
2. Port number used is 8765

What will be the output when a HTTP POST request is made to the above controller with url as <http://localhost:8765/infy> and below data?

```
{  
    "employeeId": 1,  
    "emailId": "andrew@infy.com",  
    "employeeName": "Andrew Wyatt",  
    "designation": "Senior Associate"  
}
```



- EmployeeDTO [employeeId=null, emailId=null, name=null, designation=null]
- EmployeeDTO [employeeId=1, emailId=andrew@infy.com, name=null, designation=Senior Associate]
- EmployeeDTO [employeeId=1, emailId= andrew@infy.com, name=Andrew Wyatt, designation= Senior Associate]
- 404 NOT\_FOUND error

Reset

Save

### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
list.add(c1);
list.add(c2);
list.add(c3);
list.add(c4);
if (list.contains(c3)) {
    list.set(1, list.get(3));
}
Stream<Trainee> stream = list.stream().filter(trainee -> trainee.getTraineeName().length() > 10);
stream.forEach((trainee) -> System.out.print(trainee.getTraineeId() + " "));
```

- 110 113
- 110 112 113
- 110 112
- 110 112 113 115

Reset

Save

### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
Customer cust2 = new Customer("Joe");
Customer cust3 = new Customer("Sam");
List<Customer> list = new ArrayList<Customer>();
list.add(cust1);
list.add(cust2);
list.add(cust3);
System.out.print("Before: " + list);
for (int j = 0; j < list.size(); j++) {
    Customer e = list.get(j);
    if (e.getCustName().equals(cust1.getCustName()))
        list.remove(j+1);
}
System.out.println(" After: " + list);
}
```

- Before: [John, Joe, Sam] After: [John, Sam]
- Before: [John, Joe] After: [John]
- Before: [John, Sam] After: [Sam]
- Before: [John, Sam] After: [Joe]

C

### Focus Area 1

- 1
- 2
- 3
- 4
- 5

```
        withdrawl(withdrawlAmount));
        Assertions.assertEquals("Insufficient Balance",exception.getMessage());
    }
}
@Test
public void validAmount()throws Exception{
    double withdrawlAmount=1000.00;
    Assertions.assertFalse(withdrawl(withdrawlAmount).equals("Your balance: "+(balance)));
}
```

What will be the output?

- invalidAmount() test case: Passes, validAmount() test case: Fails
- invalidAmount() test case: Passes, validAmount() test case: Passes
- invalidAmount() test case: Fails, validAmount() test case: Fails
- invalidAmount() test case: Fails, validAmount() test case: Passes

Reset

Save

```
for (String element : set) {  
    System.out.print(element + " ");  
}
```

Assumption:

1. All necessary configuration and imports are done

Note: The elements might appear in un-ordered fashion

- Earth Mars Jupiter Saturn
- Earth Mars Saturn
- Saturn Jupiter Mars Earth
- Saturn Jupiter Earth

Reset

Save

F

Focus Area 1

- 1
- 2
- 3
- 4
- 5

Question 1

What will be the output of the code given below? Note: Consider all necessary imports are added.

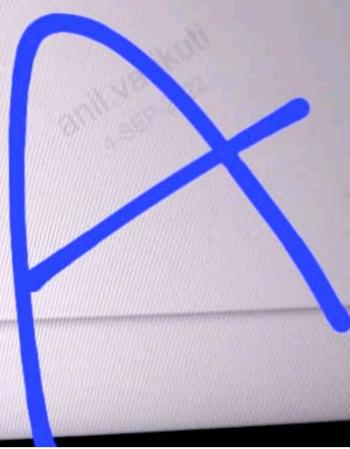
```
public static void main(String[] args) {
    LocalTime time = LocalTime.of(22, 58, 15);
    time.plusMinutes(5);
    time=time.plusHours(1);
    System.out.println("Time-"+time);
}
```

38% 62%

- Time-23:03:15
- Time-23:58:15
- Time-00:03:15
- Time-22:58:15

Reset

Save



```
public class ValidatorClassTest{  
    @Test  
    public void validatePasswordTest(){  
        // Line1  
    }  
}
```

Which of the following assertion statements would be appropriate at 'Line1' for valid test cases. Choose TWO correct answers.

- Assertions.assertEquals(Boolean.TRUE,Validator.validatePassword("Abc123")); 
- Assertions.assertTrue(Validator.validatePassword("Abc")); 
- Assertions.assertEquals(TRUE,Validator.validatePassword("Abc123")); 
- Assertions.assertTrue(Validator.validatePassword("Abc1")); 