

## Spring Associate Certification MCQs

1. Assumption: All necessary imports are done.
  - ☐ C
  - ☐ D
  - ☐ B
  - ☒ `ConcurrentModificationException` exception will be thrown. - V
  
2. Assumption: All necessary imports are done.
  - ☐ 101- Blowing with the wind  
102 – Stand by Me  
Null – Annie’s Song  
104 – Here I Am
  - ☐ 101- Blowing with the wind  
102 – Stand by Me
  - ☒ `NullPointerException` exception will be thrown - V
  - ☐ Null – Annie’s Song  
101- Blowing with the wind  
102 – Stand by Me  
104 – Here I Am
  
3. Assuming the below code executes in the main method. What will be the output of following code?
  - ☐ 2013-03-02
  - ☒ 2013-02-28 - V
  - ☐ 2013-03-01
  - ☐ Error: Cannot add month
  
4. Output?
  - ☐ 9001
  - ☒ 9001 9003
  - ☐ 9001 9002 9003
  - ☐ 9002 9003
  
5. Note: Assume the required Imports are added.
  - ☐ Peter 1001
  - ☒ Jack 1002 - V
  - ☐ Jack Peter
  - ☐ Jack Peter 1002
  
6. Output?
  - ☒ Before: [John, Joe, Sam] After:[Joe, Sam] - V
  - ☐ Before: [John, Joe] After:[Joe]
  - ☐ Before: [John, Joe, Sam] After:[John, Joe]
  - ☐ Before: [John, Sam] After:[Joe]

## Spring Associate Certification MCQs

7. Assuming the below code executes in the main method. What will be the output of following code?
- ☐ 2021-09-09
  - ☒ 2022-08-09 - V
  - ☐ 2022-09-09
  - ☐ 2022-08-02
8. Assumption: All necessary imports are done
- ☐ [5000.0,56001.4]
  - ☐ [28000.7]
  - ☐ [2500.0,28000.7]
  - ☒ IllegalStateException exception will be thrown - V
9. Assumption: All necessary imports are done
- ☐ SATURDAY
  - ☒ SUNDAY - V
  - ☐ DateTimeException exception is thrown.
  - ☐ IndexOutOfBoundsException exception is thrown.
10. Note: Assume the required imports are added.
- ☐ {1002 = Annie}
  - ☒ {1002 = Annie, 1010=joey}
  - ☐ Error: ArrayIndexOutOfBoundsException
  - ☐ {1001=Chandler,1002 = Annie}
11. Output?
- ☒ 110 113 - V
  - ☐ 110 112 113
  - ☐ 110 112
  - ☐ 110 112 113 115
12. Assumption: All necessary imports are done
- ☒ {null = null, A= Avan, B= Bob, C= Catlyn} - V
  - ☐ {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.
13. Assumption: All necessary configuration imports are done
- ☐ 9:45:59
  - ☒ 9:59:45
  - ☐ 09:45:59
  - ☐ 09:59:45

## Spring Associate Certification MCQs

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

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

15. Note: Consider all necessary imports are added.

- {Spring = 0, Java = 50, spring =20}
- ~~{Java = 50, spring =10, spring=20} - V~~
- {Java = 20, Spring =10, spring =20}
- {Java =50, spring =20}

16. Assumption: All necessary imports are done

- Assertions.assertEquals(true, citizen.isVoter(18)) - V
- Assertions.assertFalse(citizen.isVoter(18))
- Assertions.assertTrue(citizen.isVoter(17))
- Assertions.assertFalse(citizen.isVoter(17)) – V

17. Choose TWO correct options.

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

18. What will be the output?

- It will delete student with studentId=2002 from STUDENT table and then throw EmptyResultDataAccessException
- It will delete student with studentId=2002 from STUDENT table and batch with batch\_id =1002 from BATCH table
- It will delete student with batch\_id=1002 from BATCH table then throw EmptyResultDataAccessException - V
- It will throw EmptyResultDataAccessException

19. What will be the output?

- True - V
- False
- IllegalStateException will be thrown
- Service.CANNOT\_BOOK\_TICKET

## Spring Associate Certification MCQs

20. What will be the following HTTP PUT requests are mapped to above code?

- <http://localhost:8080/api/products?id=1001&name=Laptop> - V
- <http://localhost:8080/api/products/id=1001/Laptop>
- <http://localhost:8080/api/products/id=1001/name=Laptop>
- <http://localhost:8080/api/products?id=1001?name=Laptop>

21. What is the option?

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

22. Choose an option.

- line 1: `@NotNull`  
line 2: `@NotNull @Pattern(regexp = "[A-Za-z]")` – V (See thoroughly)  
line 3: `@PastOrPresent`
- line 1: `@NotNull`  
line 2: `@Pattern(regexp = "[A-Za-z]")`  
line 3: `@PastOrPresent`
- line 1: `@NotNull`  
line 2: `@Pattern(regexp = "[A-Za-z]")`  
line 3: `@Past`
- line 1: `@NotNull`  
line 2: `@NotNull @Pattern(regexp = "[A-Za-z]")`  
line 3: `@Past`

23. Choose Options.

- `@Query("select p from Product p where p.manufactureedDate > ")`  
`List<Product> findByDate(LocateDate date)`
- `@Query("select p from Product p where p.manufactureedDate > :mdate ")`  
`List<Product> findByDate(LocateDate date)`
- `@Query("select p from Product p where p.manufactureedDate > mdate ")`  
`List<Product> findByDate(@Param("mdate") LocalDate mdate)`
- `List<Product> findByManufacturedDateAfter(LocateDate date)` - V

24. Choose an option.

- Order placed successfully
- Code will run but nothing will be displayed
- `NoSuchBeanDefinitionException` exception will be thrown - V
- None of the options

25. Choose from the below options to execute the test cases successfully

- `assertThrows, assertEquals` - V
- `assertNotEquals, assertThrows`
- `assertTrue, assertThrows`
- `assertThrows, assertFalse`

## Spring Associate Certification MCQs

26. What will be the value of pat\_id in ig\_gen table after....
- 110
  - 117
  - 103 - V
  - 115
27. What will be the output.
- Loan deleted Successfully for : 1001
  - Null
  - `illegalStateException` will be thrown - V
  - Loan deleted Successfully for: null
28. Output?
- String url = "<http://localhost:5555/citizens/citizenType>";
  - String url = "<http://localhost:5555/{citizenType}>";
  - String url = "<http://localhost:5555/citizens/{citizenType}>"; - V
  - String url = "</citizens/{citizenType}>";
29. All the necessary imports and configurations are done.
- `@Query` should be used instead of `@NamedQuery`
  - The value of nature attribute in `@NamedQuery` is correct
  - In repository `@Param` annotations is not required - V
  - The value of name attribute in `@NamedQuery` is incorrect
30. Which Bean validation....
- Line1: `@Min(value = 10000) @Max(value = 99999)`
  - Line2: `@Pattern(regex = "[a-z][A-Z]+")`
  - Line3: `@Past`
  - Line4: `@NotNull` - V
31. What will be the following HTTP GET request can be handled by the above controller?
- <http://localhost:1111/api/events/venue=C4-Hall>
  - <http://localhost:1111/events/venue=C4-Hall>
  - <http://localhost:1111/events/C4-Hall>
  - <http://localhost:1111/api/events/C4-Hall> - V
32. What will be the value of primary key generated when the new customer is added?
- 1001
  - 1000
  - 2000 - V
  - 1
33. All necessary imports are done
- `restTemplate.deleteForObject(url, studentDTO.class, studentId);`

## Spring Associate Certification MCQs

- `restTemplate.delete(url, studentDTO.class, studentId);`
  - `restTemplate.delete(url, studentId);` - V
  - `restTemplate.deleteForObject(url, studentId);`
34. What methods will match the following pointcut expression?
- **None:SpringAOP supports only advising public methods.** - V
  - All Private methods of `com.infy.Allocation` class whose name is “find”, takes any number of arguments and returns void
  - Any Private methods of `com.infy.Allocation` class whose name is “find”, takes any number of arguments
  - Private methods of `com.infy.Allocation` class whose name is “find”, takes any number of arguments
35. Choose an option
- Welcome to Spring Book  
Hello World  
I am a Spring Boot Application – V
  - Welcome to Demo Application  
Hello World  
I am a Spring Boot Application
  - **I am a Spring Boot Application  
Welcome to Demo Application  
Hello World – OR 1<sup>st</sup> option(not Sure)**
  - Exception generated at the Line 1 due to an ambiguity in the key “Welcome”
36. Choose an option.
- Add following annotations to path variable `employeeId`;  
`@Min(value=1000)`  
`@Max(value=9999)`
  - **Add following annotations to path variable `employeeId`;  
`@Valid`  
`@Min(value=1000)`  
`@Max(value=9999)` - V**
  - Add following annotations to path variable `employeeId`;  
`@Min(value=1000)`  
`@Max(value=9999)`  
Add `@Validated` annotation at class level
  - Add following annotations to path variable `employeeId`;  
`@Min(value=1000)`  
Add `@Validated` annotation at class level.

## Spring Associate Certification MCQs

37. Choose TWO correct options.

- ☐ @OneToMany(cascade = CascadeType.ALL)  
@JoinColumn(name = "stream\_id")
- ☐ @ManyToOne(cascade = CascadeType.ALL)  
@JoinColumn(name = "stream\_id")
- ☐ @OneToMany(cascade = CascadeType.ALL)  
@JoinColumn(name = "streamid")
- ☐ @OneToMany(cascade = CascadeType.ALL)  
@JoinColumn(name = "stream\_id", unique = true) - V

38. Choose an option

- ☐ Bank

bank_id	bank_name	bank_branch
1001	Axis Bank	Mysore DC
1002	ICICI Bank	Bangalore DC
1003	State Bank of India	Mysore DC
1005	Indian Bank	Chandigarh DC

39. What is output?

- ☐ Name = Monica  
City = null - V
- ☐ Name = Monica  
City = Washington
- ☐ NoSuchBeanDefinitionException will be thrown
- ☐ Name = Monica will be displayed and NullPointerException will be thrown

40. What will be the value of productId generated, when a new row is inserted.

- ☐ 2003 - V
- ☐ 1
- ☐ 0001
- ☐ 2000

41. Assumption: All necessary configuration and imports are done.

- ☐ Earth Mars Jupiter Saturn
- ☐ Earth Mars Saturn - V

42. Assumption: All necessary configuration and imports are done

- ☐ [36, 36, 18, 78, 480]
- ☐ [18, 36, 36, 78, 240]
- ☐ [18, 36, 18, 78, 240]
- ☐ [18, 36, 18, 78, 480] – V

43. Choose appropriate option that satisfies the above requirements

- ☐ Arrays.asList(objects).stream().map(obj -> obj.toLowerCase()).filter(obj -> obj.charAt(0)=="i").sorted((obj1, obj2) -> obj1.length()-obj2.length()).forEach(obj -> System.out.println(obj)); - V

## Spring Associate Certification MCQs

- `Arrays.asList(objects).stream().filter(obj -> obj.startsWith("I")).sorted((obj1, obj2) -> obj2.length()-obj1.length()).forEach(obj -> System.out.println(obj));`
  - `Arrays.asList(objects).stream().filter(obj -> obj.matches("[LI].*")).sorted((obj1, obj2) -> obj1.length()-obj2.length()).forEach(obj -> System.out.println(obj));`
  - `Arrays.asList(objects).stream().filter(obj -> obj.charAt(0) == "I" || obj.charAt(0) == 'L' ).sorted().forEach(obj -> System.out.println(obj));`
44. Assume all imports are done.
- `List<Trainee> findByDateOfJoiningBetween(LocalDate fromDate, LocalDate toDate)`
  - `List<Trainee> findByDateOfGreterThanAndLessThan(LocalDate greaterThanDate, LocalDate lessThanDate) - V`
  - `List<Trainee> findByDateOfJoiningInBetween(LocalDate fromDate, LocalDate toDate)`
  - `List<Trainee> getByDateOfJoiningBetween(LocalDate fromDate, LocalDate toDate)`
45. Output?
- `Service.INCORRECT_BONUS`
  - Your Performance was satisfactory  
Your Performance was satisfactory
  - Your Performance was satisfactory
  - `Your Performance was excellent`  
`Your Performance was satisfactory - V`
46. Which of the following code snippet can be used to hit the getUsers() method...
- `String url = http://localhost:8765/users/users;  
List<UserDTO>userDTOs=restTemplate.getForObject(url,List<UserDTO>.class);`
  - `String url = http://localhost:8765/users/users;  
List<UserDTO>userDTOs=restTemplate.getForObject(url,List.class);`
  - `String url = http://localhost:8765/users;  
List<UserDTO>userDTOs=restTemplate.getForObject(url,List.class);`
  - `String url = http://localhost:8765/users/users;  
UserDTO userDTOs = restTemplate.getForObject(url, UserDTO.class); - V`
47. 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



## Spring Associate Certification MCQs

48. All necessary configurations and imports are done
- `@AfterThrowing(pointcut = "execution(* .com.infy.service.*Impl.*(..))"`
  - `@AfterThrowing(pointcut = "execution(* .com.infy.service.*Impl(..)",`  
`throwing = "exception")`
  - `@AfterThrowing(pointcut = "execution(* .com.infy.service.*Impl.*(..)",`  
`throwing = "a")`
  - `@AfterThrowing(pointcut = "execution(* .com.infy.serv.*Impl.*(..))",`  
`throwing = "exception")` – V (check thoroughly)