Question - 1

SCORE: **50 points**

Spring Boot: Rest Controller Advice Setup

Spring Boot   REST Controller Advice   Easy

The `@ControllerAdvice` annotation in Spring Boot is used as an interceptor to handle exceptions thrown from the controller's `@RequestMapping` methods. It acts as a global exception-handling component for the entire application. This question is about setting up one such controller advice component using the `@ControllerAdvice` and `@ExceptionHandler` annotations.

The definitions and detailed requirements are listed as follows.

Given a rest controller class `FizzBuzzController.java`, there is a `GET` API endpoint to be modified, which can throw three types of runtime exceptions:

1. FizzException
2. BuzzException
3. FizzBuzzException

Below is the endpoint for the `GET` API with a single path parameter:

`GET /controller_advice/{code}`
Exceptions are to be thrown based on the value of the path param {code} passed to the REST API.

Here is a series of requests and their corresponding expected responses:

`GET` request to `/controller_advice/fizz`:
- Should trigger the `FizzException`
- Should return the response code 500 and the `GlobalError` as the response body

`GET` request to `/controller_advice/buzz`:
- Should trigger the `BuzzException`
- Should return the response code 400 and the `GlobalError` as the response body

`GET` request to `/controller_advice/fizzbuzz`:
- Should trigger the `FizzBuzzException`
- Should return the response code 507 and the `GlobalError` as the response body

`GET` request to `/controller_advice/success`:
- Should not trigger any exceptions
- Should return the response code 200 and the `FizzBuzzResponse` as the response body

Complete the project so that it passes all the test cases when running the provided unit tests. To do so, implement a controller advice global exception handler `FizzBuzzExceptionHandler`, intercept the runtime exceptions thrown from the `GET` API endpoint, and return a response entity wrapped in the `GlobalError` class.

The runtime exceptions and error response classes are already predefined in the project. Complete the implementation controller `GET` API endpoint and the controller advice classes.

1/26

`GET` **request to** `/controller_advice/fizz`

The response code is 500, and when converted to JSON, the response body is:

```
{
    "message": "Fizz Exception has been thrown",
    "errorReason" : "Internal Server Error"
}
```

`GET` **request to** `/controller_advice/buzz`

The response code is 400, and when converted to JSON, the response body is:

```
{
    "message": "Buzz Exception has been thrown",
    "errorReason" : "Bad Request"
}
```

`GET` **request to** `/controller_advice/fizzbuzz`

The response code is 507, and when converted to JSON, the response body is:

```
{
    "message": "FizzBuzz Exception has been thrown",
    "errorReason" : "Insufficient Storage"
}
```

`GET` **request to** `/controller_advice/success`

The response code is 200, and when converted to JSON, the response body is:

```
{
    "message": "Successfully completed fizzbuzz test",
    "statusCode": "200"
}
```

Question - 2
React: Currency Converter
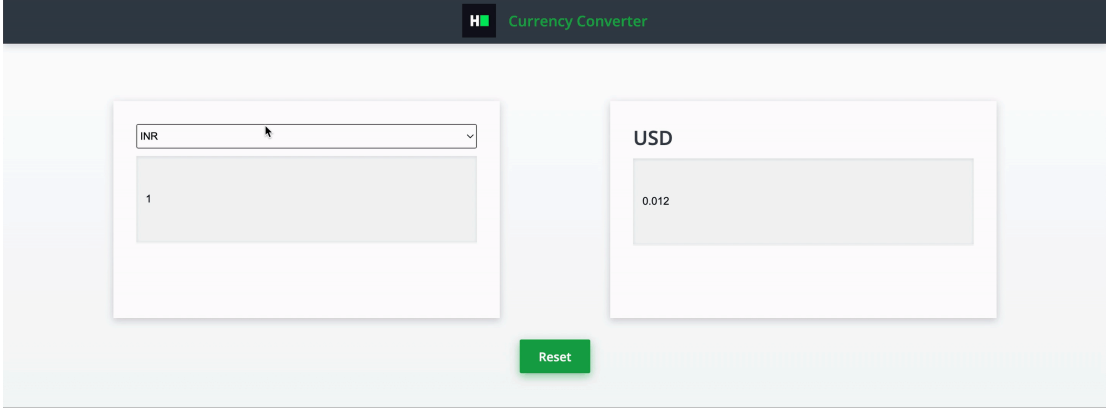
SCORE: **50 points**

React | Front-End Development | React Hooks | Easy

Complete the currency converter component that converts the value in any of the given currencies to USD as shown.

Hide animation Show animation

The list of available currencies is imported as "exchangeRates" in the component that can be used to render the list of currencies. The initial values for the currency and the input box are also imported from "exchangeRates.js".

It is given that:
- The first input box is to be used to take input.
- The second input box is read-only to display the converted value.
- Only positive integer or decimal values can be taken as input.

The component must have the following functionalities:

- The default value for:
  - input box should be 1.
  - the currency to display in the select menu should be INR.
  - The second input box should be the conversion of 1 INR to USD i.e. 0.012.
- The select menu should have five currencies in the options.
- The conversion should happen automatically:
  - when there is any input change.
  - on selecting another currency.
- The converted value should be up to 3 decimal places.
- The placeholder text should change automatically according to the currency.
- The Reset button should reset the input to 1 and give the conversion in the current currency.

The following data-testid attributes are required in the component for the tests to pass:

Component	Attribute
select menu	select-currency
first input box	input-value
second input box	converted-value
Clear Input button	clear-value

Note:
- Components have data-testid attributes for test cases and certain classes and ids for rendering purposes. They should not be changed.
- The only file that should be modified by the candidate is "src/components/currency-converter/index.js". It is open by default in the system editor.
- Avoid making changes to other files in the project structure.

Question - 3
Cryptocurrency Wallets Monitoring   SCORE: **50 points**

Databases   Easy   SQL   Aggregation   Simple Joins   Interviewer Guidelines

Write a query to identify wallets that meet two specific criteria: they have more than one transaction (at least two records with the same ID in the transactions table) and their total transaction amount exceeds 10. The output should display the wallet ID and the total amount of all transactions for that wallet.

▼ Schema

There are 2 tables:

wallets		
name	**type**	**description**
id	SMALLINT	Wallet ID
transaction_id	SMALLINT	Transaction ID

transactions		
name	**type**	**description**
id	SMALLINT	Transaction ID
amount	DECIMAL(4,2)	Transaction amount

▼ Sample Data Tables

For the sample data in tables:

wallets	
id	**transaction_id**
1	19
1	20
2	14
2	4
3	17
3	8
4	1
4	7
6	15
6	16
6	5

7	11
7	12
7	13
7	3
7	9
8	6
9	18
10	10
10	2

transactions	
id	amount
1	3.38
2	3.38
3	3.41
4	9.43
5	0.68
6	14.91
7	2.85
8	6.21
9	9.87
10	2.89
11	4.85
12	3.12
13	3.27
14	6.89
15	0.28
16	8.70
17	3.22
18	1.63
19	5.02

20	3.99

the expected output is:

wallet_id	total_amount
2	16.32
7	24.52

Question - 4
Equality

SCORE: **5 points**

`Easy` `Javascript` `Type Checking`

What is the output of the following code?

```javascript
const str = "1";
const obj = new String("1");
console.log(obj == "1");
console.log(obj === str);
console.log(obj == str);
```

- ○ True, False, False
- ● True, False, True
- ○ True, True, True
- ○ True, True, False

Question - 5
JavaScript: Typecasting Functions II

SCORE: **5 points**

`JavaScript` `Easy`

Match the JavaScript typecasting functions with their respective output.

Function Call	Output
1.*Boolean(Infinity)*	a. 20
2.*Number([20, 30])*	b. true
3. *Boolean("")*	c. NaN
4. *Number(["20"])*	d. false

○   1-b
          2-c
          3-d
          4-a

○   1-d
          2-a
          3-b
          4-c

○   1-d
          2-b
          3-a
          4-c

○   None of the above

Question - 6
JavaScript Basics

SCORE: **5 points**

`Easy` `JavaScript` `Javascript Aptitude`

JavaScript is perhaps the most widely used programming language for Web client purposes. All modern Web browsers use it. Which of the following are true?

○   JavaScript is a strongly typed language. This implies that adding a number to a string is not possible and will result in a runtime error.

◉   When JavaScript code is executed in a browser, it uses client computational resources rather than server resources.

○   JavaScript does not support recursion.

◉   Angular, React, and Vue are popular JavaScript frameworks used in Web development.

Question - 7
Declare variables with const keyword in JavaScript

SCORE: **5 points**

`Easy` `Javascript Aptitude`

One way to declare a variable in JavaScript is to use the keyword *const*. Which of the following are true about *const*? Choose one or more.

○   A variable declared with const is always globally scoped.

◉   A variable declared with const is always block-scoped.

○   A variable declared with const can be re-declared.

○   A variable declared with const can be assigned to later.

Question - 8
Closures

SCORE: **5 points**

`Javascript` `Javascript Events` `Closures` `Easy` `Closure`

What is the output of the following code?

```
const series = (...funcs) => (arg) => {
    return funcs.reduce((prev, curr) => curr(prev), arg);
};
const result = series(num => num * 10, num => num + 20, num => num / 2, num => num - 5)(4);
```

- ( ) 20
- (●) 25
- ( ) 30
- ( ) 0

Question - 9
State Management

State Management   Easy

Which change to the following code will cause it to increment *count* every second.

```
function Counter() {
 let [count, setCount] = useState(0);
 useEffect (() => {
  let id=setInterval(() => {
   setCount(count + 1);
 }, 1000);
return () => clearInterval(id);
}, []);
return <h1>{count}</h1>;
}
```

- ( ) Replace line 5 with setCount(++count);
- ( ) Replace line 5 with setCount(prevCount+ 1);
- ( ) Replace line 5 with setCount(count++);
- (●) Replace line 5 with setCount(prevCount => prevCount+ 1);
- ( ) None of the above

Question - 10
React: True statements

React   Front-End Development   Easy

Which of the following statement(s) is true about ReactJS?

1. Only one root element can be returned per component.
2. Both arguments of the useEffect hook are compulsory.
3. The useState hook returns an object.
4. The useState hook cannot be used outside of a functional component.

(●) 1

( ) 2

( ) 3

(●) 4

Question - 11
React: Children

SCORE: **5 points**

React   Front-End Development   Props   Easy

What is the output?

```
import React from 'react'

function Test (props) {
  return (
    <div>
      <p>Hello {props.name}</p>
      <p>{props.children}</p>
    </div>
  );
}
export default function App() {
  return (
    <Test name="candidate" children="Frontend">
      <p>Welcome to Hackerrank!</p>
    </Test>
  )
}
```

( ) Hello candidate
      Frontend

(●) Hello candidate
      Welcome to Hackerrank!

( ) Hello candidate
      undefined

( ) Error

Question - 12
React: Check statements

SCORE: **5 points**

```
import React from 'react';
function Sample(props) {
  return (
    <>
     <div>
        <p>{props.name}</p> {/* Statement 1 */}
        <p>{props[designation]}</p> {/* Statement 2 */}
     </div>

      <div>
        <p>{props[0]}</p> {/* Statement 3 */}
        <p>{props['designation']}</p> {/* Statement 4 */}
     </div>
    </>
    )
}
export default Sample;
```

Which of the statement(s) will not generate an error? ('name' and 'designation' are the props passed to the <Sample /> from the App component)

- ◉ Statement 1

- ○ Statement 2

- ◉ Statement 3

- ◉ Statements 4

Question - 13
React: Hooks and Their Applications   SCORE: **5 points**

React   Hooks   Easy

Match the commonly used React hooks and their applications.

Hook	Application
1. useState	a. Perform side effects on updates
2. useEffect	b. Returns memorized values
3. useContext	c. Used to track state
4. useMemo	d. Manage state globally

- ○ 1.b
     2.c
     3.d
     4.a

- ◉ 1.c
     2.a
     3.d

4.b

○   1.d
       2.c
       3.a
       4.b

○   1.b
       2.d
       3.a
       4.c

Question - 14
React: Styling in JavaScript

`React`   `CSS3`   `Easy`   `Styling React`

Select the appropriate option to generate inline styles equivalent to the CSS stylesheet.

```
.my-style{
    background-color: blue;
    color: red;
    font-size: large;
    border: solid 10px black;
}
```

○
```
const MyComponent = () => {
        return (
                <div style="background-color: blue;
                                color:red;
                                font-size:large;
                                border:solid 10px black; ">
                        My CSS Div
                </div>
                )
};
```

○
```
const MyComponent = () => {
        return (
                <div style={background-color: "blue",
                                color:"red",
                                font-size:"large",
                                border:"solid 10px black"}>
                        My CSS Div
                </div>
                )
};
```

◉
```
const MyComponent = () => {
        return (
                <div style={{background-color: "blue",
                                color:"red",
                                fontSize:"large",
```

```
                                    border: "solid 10px black"}}>
                    My CSS Div
            </div>
        )
    };
```

```
    const MyComponent = () => {
            return (
                    <div style={{'background-color': "blue",
                                    'color':"red",
                                    'fontSize':"large",
                                    'border': "solid 10px black"}>
                            My CSS Div
                    </div>
            )
        };
```

Question - 15
React: Valid Routes

React   React Router   Routes   Easy

Which route is not valid for this React Router component?

```
<Routes>
        <Route path='/' element={<Home />} />
        <Route path='/user/:id' element={<User />} />
        <Route path='/user/:id/edit' element={<UserEdit />} />
        <Route path='/post/:id' element={<Post />} />
        <Route path='/posts' element={<Posts />} />
</Routes>
```

○ /user/1/edit

○ /post/10

● /post

○ /profile/12

Question - 16
Initial Rendering

React   Easy   Functional Components   State Management

What is the output of the following code?

```
    import React, { useState } from "react";
```

```
const App = () => {
  const [number, setNumber] = useState(0);
  console.log(number);//output
  return (
    <>
      <p>{number}</p>
      <button onClick={setNumber((pre) => pre + 1)}>increment</button>
    </>
  );
};

export default App;
```

○ 0

○ 1

○ undefined

◉ Error

Question - 17
State Update

SCORE: **5 points**

React | React Hooks | Functional Components | Easy

Consider the following code.

```
import React, { useState } from "react";

const App = () => {
  const [student, setStudent] = useState({
    Name: "Abc",
    Roll: "123",
    Presence: 0,
  });

  const update = (Presence) => {
  };

  return (
    <>
      <p>{student.Presence}</p>
      <button onClick={() => update(student.Presence + 1)}>
        Update Student
      </button>
    </>
  );
};

export default App;
```

Which option increments the student presence by 1?

○ this.setState({Presence});

○ setStudent(pre=>{...pre,Presence});

◉ setStudent({ ...student, Presence });

○ None of the above

13/26

Question - 18
Method References

Java 8

```java
import java.util.*;
public class Test {
    public static void main(String[] args) {
        List<String> list = Arrays.asList("banana", "apple", "cherry");
        list.sort(String::compareToIgnoreCase);
        list.forEach(System.out::println);
    }
}
```

○ banana apple cherry

◉ apple banana cherry

○ Order is random
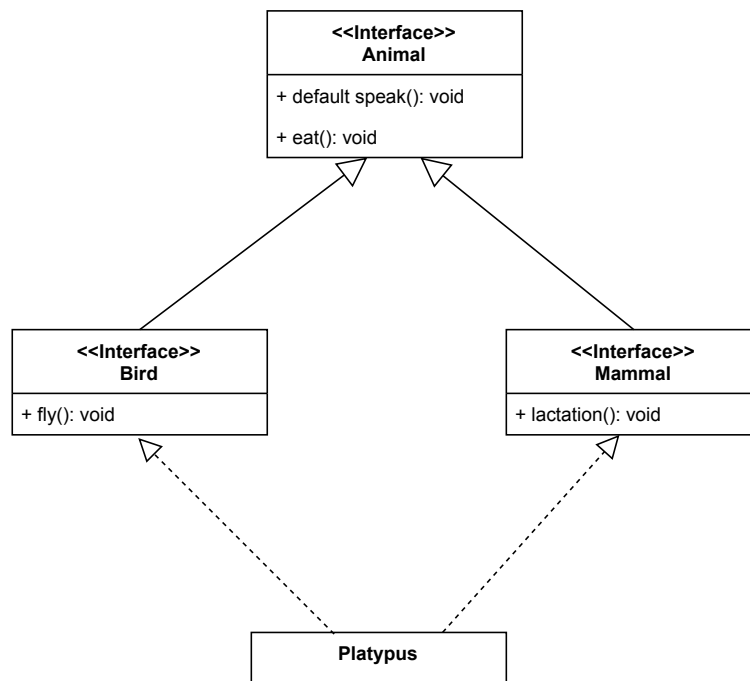
○ Compilation error

Question - 19
Interfaces

Interfaces   Classes   Medium   Inheritance

Consider the following diagram

Which of the following statements are true regarding it?

○ The Platypus class must implement all the methods.

○ The diagram represents a diamond problem.

◉ The Platypus class only have to implement the eat(), fly(), and lactation() method.

○ The Platypus class cannot extend multiple interfaces.

Question - 20
Java Garbage Collection

Garbage Collection   Medium

Which of the following statements are true about garbage collection in Java?

◉ The garbage collector can only free memory that was allocated using the new keyword.

○ The garbage collector can free memory that is being referenced by a static variable.

○ The system.gc() method can be used to guarantee that garbage collection will occur.

◉ The finalize() method can be used to prevent an object from being garbage collected.

Question - 21
OOPs

Java

A base class defines withdraw() method. Savings account applies withdrawal limit, while checking account allows overdraft up to a limit.

Question:
What OOP technique can you use here to ensure the base withdraw() behavior is customized safely by subclasses?

○ Method overloading

◉ Method overriding with super call when needed

○ Static method inheritance

○ Composition over inheritance

Question - 22
Best Three Scores

Java   Java 8   Stream   Medium

int [] scores = {1,3,8,3,5,6,2,4}

Which code returns the largest three numbers from the *scores* array?

◉ List<Integer>bestScore = Arrays .stream(scores) .boxed() .sorted() .skip(5) .collect(Collectors.toList());

○ List<Integer>bestScore = Arrays .stream(scores) .boxed() .sorted() .skip(3) .collect(Collectors.toList());

○ List<Integer>bestScore = Arrays .stream(scores) .sorted() .skip(5) .collect(Collectors.toList());

○ List<Integer>bestScore = IntStream.of(scores) .sorted() .boxed() .skip(4) .collect(Collectors.toList());

Question - 23
Streams API

`Java 8`

```java
import java.util.*;
import java.util.stream.*;
public class Test {
    public static void main(String[] args) {
        List<Integer> nums = Arrays.asList(1,2,3,4,5,6);
        int sum = nums.stream()
                .filter(n -> n % 2 == 0)
                .mapToInt(n -> n * 2)
                .sum();
        System.out.println(sum);
    }
}
```

● 24

○ 12

○ 20

○ Compilation error

Question - 24
Optional Class

`Java 8`

```java
import java.util.Optional;
public class Test {
    public static void main(String[] args) {
        Optional<String> opt = Optional.ofNullable(null);
        System.out.println(opt.orElse("Default"));
    }
}
```

○ null

● Default

○ Empty

○ Compilation error

Java   Java 8   Stream   Medium

Consider the following code:

```
int sum  = Stream.iterate(new int[]{0, 1},
tint sum  = Stream.iterate(new int[]{0, 1}, x -> new int[]{x[1], x[0] + x[1]})
                   .limit(n)
                   .map(x -> x[0])
                   .collect(toList())
                   .stream()
                   .distinct()
                   .filter(i ->  i % 2 == 0)
                   .mapToInt(i->i).
                   sum();
return sum;
```

What is the difference in results for *n=7* and *n=8*?

- ○ 2
- ○ 4
- ○ 8
- ● 0
- ○ 11

MySQL   Medium

Select the expression that causes a MySQL error.
Select all that apply.

- ○ SELECT AVG(id) FROM categories
- ○ SELECT AVG(id) FROM categories GROUP BY type
- ● SELECT 'average', AVG(id) FROM categories GROUP BY 1
- ○ None of the above, all expressions are correct

MySQL   Medium

Choose the expression that retrieves the base domain "domain.com".

- (●) SELECT SUBSTRING_INDEX('my.subdomain.domain.com', '.', -2)

- (○) SELECT SUBSTRING_INDEX('my.subdomain.domain.com', 'my.subdomain.', 1)

- (○) SELECT SUBSTRING_INDEX('my.subdomain.domain.com', '.', 2, 2)

- (○) All expressions are correct

Question - 28
MySQL: DEFAULT Constraint

MySQL   Medium

Select the expressions that cause a MySQL error.

- (○) CREATE TABLE users (user VARCHAR(255) DEFAULT 'test')

- (○) CREATE TABLE users (status VARCHAR(255) DEFAULT NULL)

- (●) CREATE TABLE users (id VARCHAR(255) DEFAULT NOT NULL)

- (●) CREATE TABLE users (id VARCHAR(255) DEFAULT RAND())

Question - 29
MySQL: UNIQUE Constraint

MySQL   Medium

Which expression causes a MySQL error?

- (○) CREATE TABLE users (first_name VARCHAR(255), last_name VARCHAR(255), CONSTRAINT unique_name UNIQUE (first_name, last_name))

- (●) CREATE TABLE users (first_name VARCHAR(255) UNIQUE, last_name VARCHAR(255) UNIQUE)

- (○) CREATE TABLE users (first_name VARCHAR(255), last_name VARCHAR(255), UNIQUE (first_name, last_name))

- (○) None of the above, all expressions are correct.

Question - 30
Spring Data Repository Type

Spring Data   Spring Boot   Medium

Consider the following code.

Admin.java

18/26

```
@Entity
@SequenceGenerator(name = "default_gen", sequenceName = "SEQ_ADMIN")
public class Admin {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "default_gen")
    @Column(name = "id")
    private String name;
    private String password;
    private String locale;
    private String email;
    //getter & setters
}
```

Admin Repository.java

```
@Repository
public interface AdminRepository extends X <Admin, Long> {

    Page<Admin> findAllByLocaleOrderByName(String locale, Pageable pageable);
}
```

To run the *findAllByLocaleOrderByName* method with pagination ability, which of the following options can be used in place of *X* in the *AdminRepository* interface?

- ◯ CrudRepository

- ⬤ JpaRepository

- ◯ PagingRepository

- ⬤ PagingAndSortingRepository

Question - 31
Spring Bean Initialization

Java   Spring   Medium   Exception Handling

What will be the output of the following code?
Vehicle.java

```
public class Vehicle
{
    public String name;

    public String getName()
    {
      return name;
    }

    public void setName(String name)
    {
      this.name = name;
    }
}
```

BeanConfiguration.java

```
@Configuration
public class BeanConfiguration
```

```
    {
        @Bean
        Vehicle vehicle1()
        {
            var veh = new Vehicle();
            veh.setName("Honda");
            return veh;
        }

        @Bean
        Vehicle vehicle2()
        {
            var veh2 = new Vehicle();
            veh2.setName("Volvo");
            return veh2;
        }
    }
```

Example.java

```
@Component
@Slf4j
public class Example implements CommandLineRunner
{
    @Override
    public void run(String... args) throws Exception
    {
        var context = new AnnotationConfigApplicationContext(BeanConfiguration.class);
        Vehicle vehicle = context.getBean(Vehicle.class);
        log.info("Vehicle name: {}", vehicle.getName());
    }
}
```

○ Volvo

○ Honda

○ Honda
  Volvo

◉ throws org.springframework.beans.factory.NoUniqueBeanDefinitionException on runtime

Question - 32
Spring Data Repository

SCORE: **5 points**

Spring Boot   Medium   Spring Data

To use Spring data repositories within another jar module and to use them in a Spring Boot application, what annotation should be used in Spring Boot configuration/ SpringBootApplication classes?

◉ @EnableJpaRepositories(basePackages = {repsitoryPackageOfJarModule})

○ @EnableRepositories(basePackages = {repsitoryPackageOfJarModule})

○ @EnableEntities(basePackages = {repsitoryPackageOfJarModule})

○ @EnableHibernateRepositories(basePackages = {repsitoryPackageOfJarModule})

Question - 33
Spring Boot Localization

Spring Boot   Medium

Which of the following options can be used for internationalization in Spring Boot?

- ⦿ MessageResource
- ◯ ParameterResource
- ⦿ ResourceBundle
- ◯ None of the above

Question - 34
Spring IOC Container

Spring   Medium

Which of the following options act as a Spring IOC Container?

- ⦿ ApplicationContext
- ◯ DispatcherServlet
- ◯ SpringRunner
- ⦿ BeanFactory
- ◯ None of the above

Question - 35
Spring Scopes

Medium   Spring

Which of the following statements is/are true about @Scope("request") and @Scope("prototype") annotations?

- ⦿ Both are used to define the scope of spring beans.
- ⦿ @Scope("request") creates a bean instance for a single HTTP request.
- ⦿ @RequestScope can be used instead of @Scope("request").
- ⦿ @Scope("prototype") will return a different instance every time it is requested from the container.

Question - 36
Autowiring Conflict

```
@Service
class ServiceA {}
@Service
class ServiceB {}
@RestController
public class TestController {
    @Autowired
    private ServiceA service;
}
```

What happens when Spring Boot runs?

○ Application fails due to ambiguity

● Application runs successfully

○ Compilation error

○ Circular dependency

Question - 37
Bean Scope

SCORE: **5 points**

```
@Configuration
public class AppConfig {
    @Bean
    @Scope("prototype")
    public StringBuilder stringBuilder() {
        return new StringBuilder("Spring");
    }
}
```

If two beans are requested from the context, what happens?

○ Both beans point to the same object

● Two different objects are created

○ Compilation error

○ NullPointerException

Question - 38
Git Data Structure

SCORE: **5 points**

Which data structure is git modelled after? In other words, when you look under the hood, which data structure does git resemble?

○ a queue

○ a linked list, every commit links to its parent

○ a stack

◉ a hash or a key-value store

○ an array

Question - 39
Comprehensive Cloud Monitoring and Performance Strategy

SCORE: 5 points

`Easy` `Cloud Monitoring and Management` `Cloud Architecture Patterns and Best Practices`

A rapidly growing Software as a Service (SaaS) company needs a cloud monitoring strategy for high availability, performance, security, and cost management.

Which option offers comprehensive monitoring across infrastructure, application, and security, with features for cost optimization?

○ Choose unconnected tools for different monitoring needs, manually check logs, use basic email alerts, adjust resources based on periodic reviews, do annual security audits, and conduct performance testing reactively.

○ Use one APM tool for infrastructure monitoring, rely on default logging, respond to user-reported issues without proactive alerts, skip autoscaling, and do frequent security checks manually.

◉ Use an integrated cloud monitoring platform like Datadog or New Relic, that offer infrastructure and application performance monitoring, use ELK Stack for log management, set up platform-based alerts, apply autoscaling and capacity planning tools, use cloud security tools, and perform tests with Apache JMeter.

○ Implement an integrated monitoring solution for infrastructure and APM, use cloud tools for log management, create comprehensive alerts for performance, security, use auto scaling and capacity planning, continuously monitor security, and systematically test performance.

Question - 40
Deployment Across Multiple Clouds

SCORE: 5 points

`Easy` `Multi-Cloud` `Container Platforms and Serverless Computing`

A financial company is building a secure online banking application for deployment across multiple clouds to ensure high availability, disaster recovery, and compliance. The application needs to handle transactions securely, protect customer data, and integrate with on-premises systems.

Which approach leverages containerization for portability and Infrastructure as Code (IaC) for automated deployments, while ensuring secure communication and data encryption?

○ Containerize with Kubernetes, use VPNs for inter-cloud communication, use TLS for data encryption, and have manual deployment strategies based on predictable usage patterns.

◉ Use Docker for containerization, implement communication through public APIs with OAuth for security, apply AES-256 encryption for data security, and use Terraform scripts for IaC to manage deployment.

Use serverless computing models across clouds without containerization, use API Gateways for communication, use database encryption using cloud solutions, and use on tools like ARM, Cloud Formation, CDM for automated deployment.

Use Docker for containerization, communication using service meshes like Istio, use mutual TLS for encryption, and use Ansible playbooks for automated deployment.

Question - 41
JUnit Multiple When Usage

SCORE: 5 points

`Java`   `JUnit`   `Mockito`   `Easy`

Consider the following code block

```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import java.util.ArrayList;
import java.util.List;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.mockito.Mockito.*;

public class JUnitWhenMultipleUsage {

    private List<String> nameList;

    @BeforeEach
    public void init() {
        nameList = mock(ArrayList.class);
    }

    @Test
    public void testWithMockedArrayList() {
        when(nameList.size()).thenReturn(10).thenReturn(20).thenReturn(30).thenReturn(40).thenReturn(41);
        assertEquals(X, nameList.size());
        assertEquals(Y, nameList.size());
        assertEquals(Z, nameList.size());

        verify(nameList, T).size();

    }
}
```

Which of the following *X, Y, Z,* and *T* values, when applied, allow the *testWithMockedArrayList* method to pass the test?

○ X = 41
   Y = 41
   Z= 41
   T = atLeastOnce()

○ X = 41
   Y = 41
   Z= 41
   T = times(3)

◉

24/26

X = 10
Y = 20
Z= 30
T = times(3)

○   X = 41
        Y = 41
        Z= 41
        T = atLeast(3)

Question - 42
Mockito Mock-Spy

SCORE: **5 points**

JUnit   Mockito   Medium

Which of the following statements is true about Mockito's mock and spy?

○   While mock creates objects that simulate the behavior of real objects, spy creates a real instance of that class.

○   With spy you can change stubbing specific methods of a given class.

○   If you create a mock of an interface and try to configure a stub to call a real method, Mockito will throw an exception.
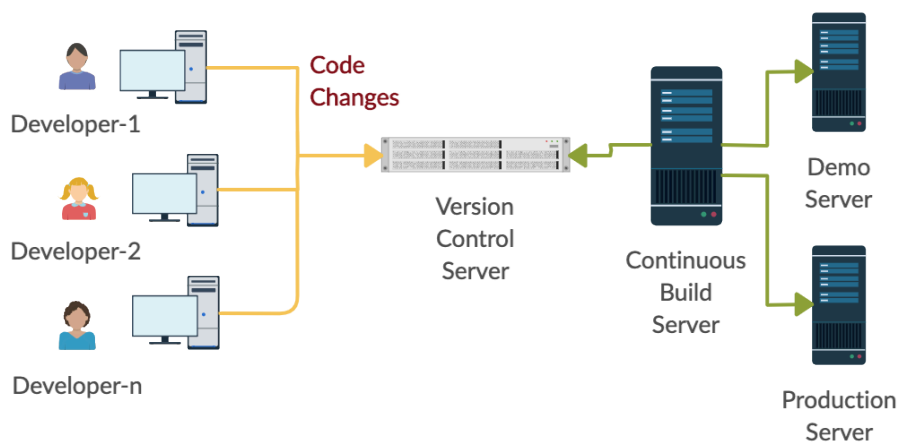
●   All of the above

Question - 43
Jenkins: Requirements

SCORE: **5 points**

Jenkins   Easy

This is an architectural diagram showing a typical CI system.

What is the basic requirement for using Jenkins?

○ Source code repository such as Git repository

○ Build script such as Maven script

● Both A and B

○ None of the above

○ Source code repository such as Git repository

○ Build script such as Maven script

● Both A and B

○ None of the above