

Quiz Questions

(1) Purpose of using the Model-view-controller

Classroom

MVC Core
Purpose of using the Model-View-Controller

Problem: Purpose of using the Model-View-Controller
Easy • Score: 20/20

Problem statement
What is the purpose of using the Model-View-Controller (MVC) architecture in Spring Boot?

Options: Pick one correct answer from below

- ☐ To separate the user interface from the business logic
- ☐ To improve the performance of the application
- ☐ To make the code more maintainable and scalable
- ☒ All of the above

Solution description
The Model-View-Controller (MVC) architecture separates the application logic into three interconnected components: the model, the view, and the controller. In Spring Boot, MVC creates web applications by providing tools to automatically configure and wire together the Model, View, and Controller components, making it easier to build and maintain a web application.

(2) Complete MVC architecture

Classroom

MVC Core
MVC architecture

Problem: Complete MVC architecture
Easy • Score: 20/20

Problem statement
Complete the following diagram of MVC architecture.

Diagram:

Options: Pick one correct answer from below

- ☒ A) A-Controller, B-View, C-Model
- ☐ B) A-View, B-Model, C-Controller
- ☐ C) A-Model, B-Controller, C-View
- ☐ D) None of the above.

Solution description

Submit

Ask a doubt

(3) Match MVC

← Classroom

MVC Core
Match MVC

?

V

≡

Problem Submissions Doubts

Match MVC

Easy • Score 20/20

Send feedback

Let's suppose a developer is working on an E-commerce website. Based on your understanding of the MVC architecture, choose the correct option.

a)Model	1. The website's user interface, including the layout, design, and navigation.
b)View	2. The website's data and business logic include the product catalogue, inventory, pricing, and customer information.
c)Controller	3. This layer handles the user input, such as when a customer adds an item to their shopping cart or searches for a product.

Options: Pick one correct answer from below

☐ a-1,b-2,c-3

☒ a-2,b-1,c-3

☐ c-1,b-2,a-3

☐ None of the above

Solution description

The Model-View-Controller (MVC) is a design pattern used in software engineering to separate the logic of a program into three interconnected components: Model: Responsible for managing the data of the application. It can also include logic for data validation, business rules, and data access. View: Responsible for displaying the data to the user. It receives data from the model and displays it on the screen. Controller: Responsible for handling user input and updating the model and view accordingly. It receives input from the user, updates the model, and then updates the view to reflect the changes in the model.

(4) Purpose of JSP

Problem Submissions Doubts

Purpose of a JSP

Easy • Score 20/20

Send feedback

What is the purpose of a JSP file in the MVC architecture in Spring Boot?

Options: Pick one correct answer from below

☐ To handle database connections

☐ To handle user input

☐ To handle server-side logic

☒ To handle the presentation layer (displaying data to the user)

Solution description

JavaServer Pages (JSP) files are the view component in the MVC architecture. They are responsible for displaying the data to the user. JSPs are similar to HTML files but contain Java code executed on the server side to generate the HTML sent to the client's web browser.

< Submit >

Ask a doubt

(5) Dependency is nested

← Classroom

MVC Core
Dependency is needed

?

V

≡

Problem Submissions Doubts

✓ Dependency is needed

Easy • Score 20/20

Send feedback

Problem statement

While working on the spring-boot project, the program must stop and start the server manually whenever code changes occur. Which dependency is needed to solve this problem?

Options: Pick one correct answer from below

☐ JSTL

☒ Dev-tools

☐ Jasper

☐ None of the above

Solution description

When you include this dependency in your project, it provides the following development-time features: Automatic application restart: whenever you make changes to your code and save the changes, your application will automatically restart so that you can see the changes immediately. LiveReload: it allows you to refresh the browser automatically when you make changes to your code.

(6) Configuration of "application.yml" file

← Classroom

MVC Core
Configuration of "application.yml"

?

V

≡

Problem Submissions Doubts

✓ Configuration of "application.yml"

Easy • Score 20/20

Send feedback

Problem statement

Application.yml file is needed to be configured so that prefix and suffix can be added. What would the "application.yml" file look like?

Options: Pick one correct answer from below

☒

```
spring:
  mvc:
    view:
      prefix: "WEB-INF/jsp/"
      suffix: ".jsp"
```

☐

```
spring:
  mvc:
    view:
      prefix: "WEB-INF/jsp/"
      suffix: ".jsp"
```

☐

```
spring:
  mvc:
    view:
      prefix: "web-inf/jsp/"
      suffix: ".jsp"
```

☐

```
spring:
  mvc:
    view:
      prefix: "WEB-INF/jsp/"
      suffix: ".JSP"
```

(7) Problem with web services

The screenshot shows a Classroom interface for a problem titled "Problem with web services". The problem is marked as "Easy" with a score of "20/20". The problem statement asks: "In the spring boot project, there is a problem with web services and servers. What dependency might not be added to the pom.xml file?". The options are: "Spring dev-tools", "Spring Web" (selected and marked correct), "Tomcat embed jasper", and "JSTL". The solution description states: "The 'spring-web' dependency is a part of the Spring Framework and is used to build web applications in Spring Boot. It provides several key components for building web applications, such as Spring's web MVC framework for handling requests and generating responses. Support for handling HTTP requests and responses, including request and query parameters."

Classroom

MVC Core
Problem identify

Problem Submissions Doubts

Problem with web services
Easy • Score 20/20

Problem statement
In the spring boot project, there is a problem with web services and servers. What dependency might not be added to the pom.xml file?

Options: Pick one correct answer from below

- ☐ Spring dev-tools
- ☒ Spring Web
- ☐ Tomcat embed jasper
- ☐ JSTL

Solution description
The "spring-web" dependency is a part of the Spring Framework and is used to build web applications in Spring Boot. It provides several key components for building web applications, such as Spring's web MVC framework for handling requests and generating responses. Support for handling HTTP requests and responses, including request and query parameters.

(8) What annotation Tag

The screenshot shows a Classroom interface for a problem titled "What annotation tag". The problem is marked as "Easy" with a score of "20/20". The problem statement asks: "While working on the homepage controller, what annotation tag would be used on the method for mapping such that it returns a homepage?". The options are: "@HomepageMapping", "@Controller", "@RequestMapping" (selected and marked correct), and "@ResponseBody". The solution description states: "This annotation is used to map a URL pattern to a method in a controller class. The method with @RequestMapping will handle incoming HTTP requests that match the specified URL pattern."

Classroom

MVC Core
Identify annotation tag

Problem Submissions Doubts

What annotation tag
Easy • Score 20/20

Problem statement
While working on the homepage controller, what annotation tag would be used on the method for mapping such that it returns a homepage?

Options: Pick one correct answer from below

- ☐ @HomepageMapping
- ☐ @Controller
- ☒ @RequestMapping
- ☐ @ResponseBody

Solution description
This annotation is used to map a URL pattern to a method in a controller class. The method with @RequestMapping will handle incoming HTTP requests that match the specified URL pattern.

(9) Which annotation

The screenshot shows a Classroom interface for a problem titled "Which annotation". The problem is marked as "Easy" with a score of "20/20". The problem statement asks: "Which annotation is used to declare a class as a controller in a Spring MVC application?". The options are: "@Controller" (selected and marked correct), "@RestController", "@Service", and "@Repository". The solution description states: "The @Controller annotation is a specialization of the @Component annotation, which marks a class as a Spring bean. The @Controller annotation signifies that a class functions as a controller in the Spring MVC framework."

Classroom

MVC Core
Which annotation

Problem Submissions Doubts

Which annotation
Easy • Score 20/20

Problem statement
Which annotation is used to declare a class as a controller in a Spring MVC application?

Options: Pick one correct answer from below

- ☒ @Controller
- ☐ @RestController
- ☐ @Service
- ☐ @Repository

Solution description
The @Controller annotation is a specialization of the @Component annotation, which marks a class as a Spring bean. The @Controller annotation signifies that a class functions as a controller in the Spring MVC framework.

(10) Spring MVC controller

← Classroom

MVC Core
Spring MVC Controller

Problem Submissions Doubts

✓ Spring MVC Controller

Easy • Score 20/20

Send feedback

Problem statement

Which methods would you use in a Spring MVC controller to return the "sign" JSP file when the login page is requested? (Multi-choice)

Options: One or more answers may be correct

☒

```
@RequestMapping(value = "/login")
public String login() {
    return "sign";
}
```

☐

```
@RequestMapping("/login")
public void login() {
    return "signs";
}
```

☒

```
@RequestMapping("/login")
public String login() {
    return "sign";
}
```

☐

```
@RequestMapping("/sign")
public String login() {
    return null;
}
```

(11) When developing big projects

← Classroom

MVC Core
Developing big project

Problem Submissions Doubts

✓ When developing big projects

Easy • Score 20/20

Send feedback

Problem statement

When developing big projects, it is a mandatory practice to use the Interface and its Implementation class when writing service and domain classes to_____

Options: Pick one correct answer from below

☐ Promote clear separation of concerns

☐ Promote modularity and flexibility in the design

☐ Allow for easier testing and maintenance of code

☒ All of the above

Solution description

Creating an interface before implementing actual classes in a big project promotes clear separation of concerns, modularity, and flexibility design and allows for easier testing and maintenance of the code. This separates the class behaviour from the implementation details and allows for predictable interactions between different parts of the project, which makes it easier to change or update the project in the fu

(12) Identify the annotation

← Classroom

MVC Core
Identify the Annotation

?

V

≡

Problem Submissions Doubts

Identify the annotation

Easy • Score 20/20

Send feedback

Problem statement

Consider a scenario where you are developing a Java-based web application and have reached the stage where you need to implement the main business logic of the application. Which of the following annotations would be most appropriate to use in this situation to indicate that a class is part of the service layer?

Options: Pick one correct answer from below

☐ @Bean

☒ @Service

☐ @Component

☐ @Controller

Solution description

The annotation that should be used in this scenario is @Service. The @Service annotation is used to indicate that a class belongs to the service layer of an application. It is a specialized version of the @Component annotation and marks a class as a Spring-managed service bean. This annotation allows Spring to automatically detect and register the service bean, making it available for injection into other application components.

(13) Employee Management application

← Classroom

MVC Core
Employee management application

?

V

≡

Problem Submissions Doubts

Employee management application

Easy • Score 20/20

Send feedback

Problem statement

Case: A developer is creating an employee management application. He is working on the service class for his user login page. He needs to make a method that creates a new employee taking the name, id, and designation as a parameter. He has predefined "createUser" (taking name, id, and designation as argument) return boolean and "saveUser" (taking no argument) methods. The codes are as follows:
Domain Layer:

```
***
public class Employee {
    public int id;
    public String name;
    public String designation;

    public boolean createUser(String name, int id, String designation) {
        // Implementation
    }
    public Integer saveUser() {
        // Implementation
    }
}
```

Web Layer:

```
***
@Controller
public class SignUpController {
    @RequestMapping("/signup")
    public String signup(){
        return "signup";
    }
}
```

Options: Pick one correct answer from below

☐

```
***
public boolean createUser(String name, int id, String designation){
    Integer idempCreated = employee.createUser(name, id, designation);
    employee.saveUser();
    return idempCreated;
}
```

☐

```
***
public boolean createUser(String name, int id, String designation) {
    Integer idempCreated = employee.createUser(name, id, designation);
    return idempCreated;
}
```

☒

```
***
public boolean createUser(String name, int id, String designation) {
    boolean idempCreated = employee.createUser(name, id, designation);
    employee.saveUser();
    return idempCreated;
}
```

☐ None of the above

Solution description

Service class act like a business logic Layer. The function's return type in the A option is boolean, not Integer. In the B option, We have not called the saveUser function.

(14) Purpose of @ModelAttribute

← Classroom

MVC Core
Purpose of @ModelAttribute

?

V

≡

Problem

Submissions

Doubts

✓ Purpose of @ModelAttribute

Easy • Score 20/20

Problem statement

Send feedback

What is the purpose of the @ModelAttribute annotation in Spring MVC?

Options: Pick one correct answer from below

☒ To bind JSP page details to a domain object

☐ To define the request handler method for a specific URL pattern

☐ To inject a bean into the controller

☐ To bind request data to a method argument

Solution description

The @ModelAttribute annotation in Spring MVC is used to bind JSP page details to a domain object. This means that values from a JSP form can be automatically populated into a Java object when the form is submitted. The Java object, known as the model attribute, is stored in the model map and can be accessed in the controller method handling the form submission.

(15) Choose how the controller class code

← Classroom

MVC Core
Controller class code

?

V

≡

Problem

Submissions

Doubts

✓ Choose how the controller class code

Easy • Score 20/20

Problem statement

Send feedback

Choose how the controller class code would look to display the message on the webpage "Welcome to coding ninjas." Suppose the return needs to be the "welcome" page(JSP). Then what would the code look like?(Multi-Choice)

Options: One or more answers may be correct

☒

```
***
@Controller
public class MyController {

    @RequestMapping("/welcome")
    public String welcomePage(Model model) {
        model.addAttribute("msg", "Welcome Coding ninjas!");
        return "welcome";
    }
}
```

☐

```
***
@Controller
public class MyController {

    @RequestMapping("/welcome")
    public String welcomePage(Model model) {
        Model.addAttribute("msg", "Welcome Coding ninjas!");
        return "welcome";
    }
}
```

☒

```
***
@Controller
public class MyController {

    @RequestMapping("/welcome")
    public String welcomePage(@ModelAttribute("msg") String message) {
        message = "Welcome Coding ninjas!";
        return "welcome";
    }
}
```

☐

```
***
@Controller
public class MyController {

    @RequestMapping("/welcome")
    public String welcomePage(@ModelAttribute("msg") String message) {
        message = "Welcome Coding ninjas!";
    }
}
```

(16) Create a form

← Classroom

MVC Core
Create form

?

V

≡

Problem Submissions Doubts

Create a form
Easy • Score 20/20

Send feedback

Problem statement
Create a form with the attribute(name).Domain class has already been created with these attributes, and we have "user" as the model attribute. The form should look as shown below.

Name:

What would the JSP page code look like?

Options: Pick one correct answer from below

☐

```
<@ taglib prefix="form" url="http://www.springframework.org/tags/form" >
</@>
<form:form action="Name" modelAttribute="user">
  <input type="text" path="Name"/>
</form>
</html>
```

☒

```
<@ taglib prefix="form" url="http://www.springframework.org/tags/form" >
</@>
<form:form action="Name" modelAttribute="user">
  <input type="text" path="Name"/>
</form>
</html>
```

☐

```
<html>
<form:form action="done">
  <input type="text" path="Name"/>
</form>
</html>
```

☐ None of these

Solution description

In option A we have taken the wrong model attribute name("user" was provided to us). If we consider option C, we have not implemented taglib and model attribute, which makes it wrong.

< Submit >

Ask a doubt

(17) Optional Class 1

← Classroom

MVC Core
Optional Class I

?

V

≡

Problem Submissions Doubts

Optional Class I
Easy • Score 20/20

Send feedback

Problem statement
What happens in Java if you don't use the Optional class correctly?

Options: Pick one correct answer from below

☐ Your code will run with no issues

☒ A NullPointerException will be thrown

☐ A NoSuchElementException will be thrown

☐ A compile-time error will occur

Solution description

The Optional class in Java is used to handle the absence of a value. It provides a way to represent an optional value instead of using null, which can lead to NullPointerExceptions.

(18) Optional Class 2

← Classroom

MVC Core
Optional Class II

?

V

≡

Problem Submissions Doubts

Optional Class II
Easy • Score 20/20

Send feedback

Problem statement
While writing a Java program to process user information. You want to retrieve the user's address from a database. However, the address may not exist for some users. To handle this problem/exception, what could be used in Java?

Options: Pick one correct answer from below

☒ Use the Java Optional class to handle the missing value

☐ Use an if-else statement to check if the value is present

☐ Raise an error and stop the program execution

☐ None of the above

Solution description

You can also use the Java Optional class, which allows you to handle the case of a missing value more elegantly and concisely. The Optional class provides methods for checking if a value is present and for accessing the value if it is present.

(19) Generics

← Classroom

MVC Core
Generics

?

V

Problem Submissions Doubts

✓ Generics

Easy • Score 20/20

Problem statement

[Send feedback](#)

While developing a Java application that stores and manages a collection of objects of various types, such as integers, strings, and custom objects. Which of the following approaches is the most suitable for this task?

Options: Pick one correct answer from below

☐ Creating separate classes or methods for each type of object

☐ Using an array to store the objects and using type casting to handle different types

☒ Using generics to define a single class or method that can work with objects of any type ✓

☐ Using the Object class to store the objects and using an instance of the operator to handle different types

Solution description

Generics in Java allow defining classes, interfaces, and methods that work with objects of any type rather than just a single type, making the code more flexible and reusable. A type parameter, defined using angle brackets <>, represents the type the generic will work with. This type is specified when creating an instance or calling the method. Using generics helps prevent runtime errors by catching type-related issues at compile time.