MVC AND JSP MODELS

Instructor:

⑩ **What Is a MVC?**

⑩ **Web Application MVC Pattern**

- ✓ **Model**
- ✓ **View**
- ✓ **Controller**

⑩ **JSP Model 1, 2**

Learning Goals

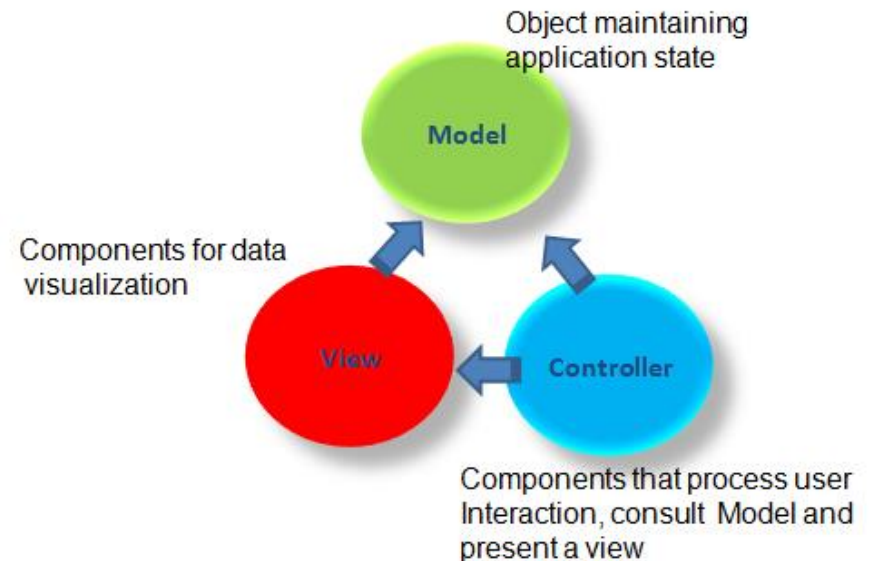**After the course, attendees will be able to:**

❖ Understand MVC - JSP Models

❖ Can use MVC – JSP Models to build Project

What Is a MVC?

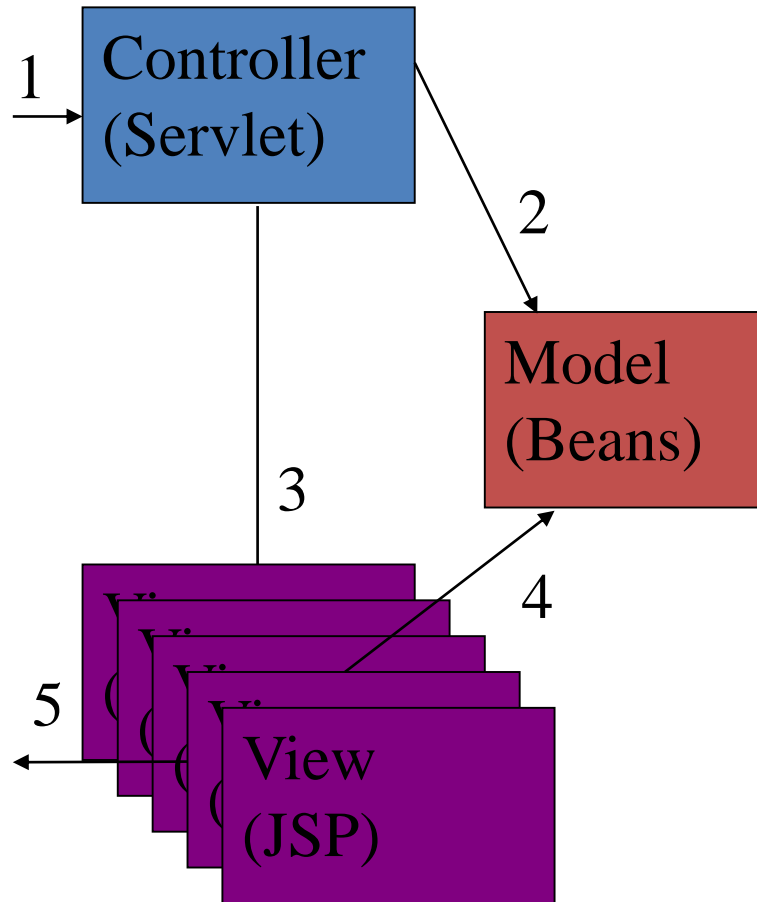❖ MVC stands for **Model** / **View** / **Controller**.

✓ A software pattern where logic is separated from the model and view in order to provide for better reuse possibilities.

✓ A software pattern recognized in the early days of small talk.

❖ MVC Architecture

Web Application MVC Pattern

Model:

- Information is provided in objects or beans

View:

- The JSP provide the view

Controller:

- Servlet provides control logic and becomes the controller

❖ The model is responsible for managing the data of the application.

    ✓ Manages Information - If Changes

    ✓ Maps Real-World Entities

❖ Contains data and Related Functionality

❖ Performing DB Queries

    ✓ Calculating Business Process

❖ Encapsulates Domain Logic which are independent of Presentation

Model
(Beans)

MVC - View

- ❖ Obtains data from model & presents to the user
- ❖ Represents Output/Input of the application
- ❖ Display results of Business Logic
- ❖ Free Access to Model
- ❖ Reads Data from Model – Using Query Methods
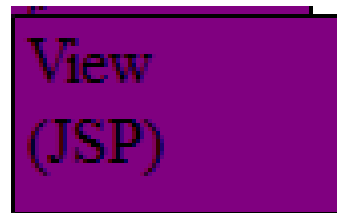
View
(JSP)

❖ Serves logical connection between user's interaction and the business process

❖ It receives and translates input to request on model or view

❖ Input from user and instructs[chỉ thị] the model and view to perform action

❖ Responsible for making decision among multiple presentation

❖ Maps the end-user action to the application response

Relationship between Components

❖ View and Controller

    ✓ Controller is responsible for creating or selecting view

❖ Model and Controller

    ✓ Controller depends on model

    ✓ If a change is made to the model then there might be required to make parallel changes in the Controller

❖ Model and View

    ✓ View depends on Model

    ✓ If a change is made to the model then there might be required to make parallel changes in the view

Logical Layers in Web Application

```
public class DbBean{
    public string userName {  get; set; }
    public string password {  get; set; }
…
}
```

Model

```
<form method="post" action="Login">
        <input type="text" name="txtUserName"></td>
        <input type="text" name="txtUserName"></td>
…
<td>${u.userName} </td>
<td>${u.userPassword} </td>
```

View

```
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)throws ServletException, IOException {
        String userName = request.getParameter("txtUserName");
        String userPassword = request.getParameter("txtPassword");
      User u = new User();
            UserBO ubo = new UserBO();
            u.setUserName(userName);
            u.setUserPassword(userPassword);…
```

Contr oller

MVC Collaboration Diagram
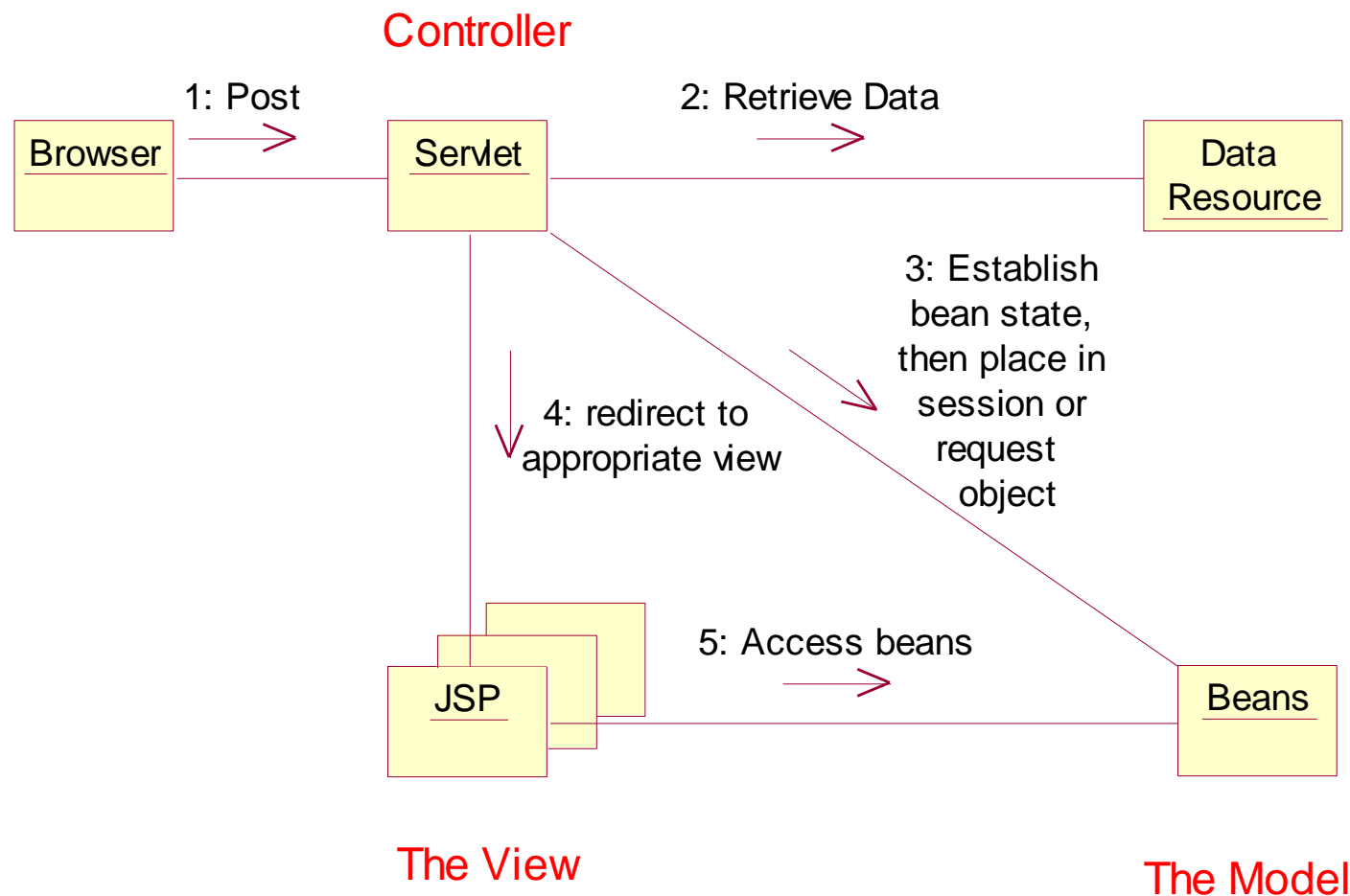
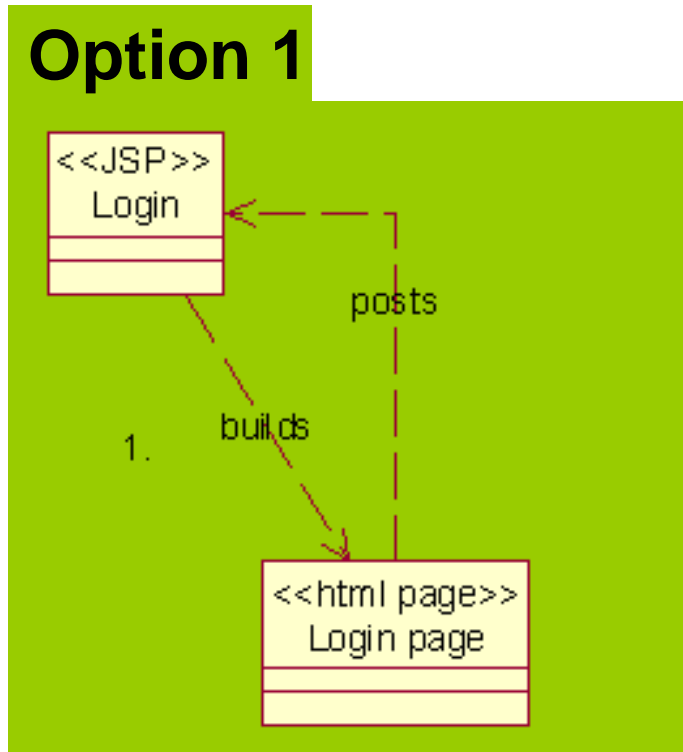# Java 2 web application options:

❖ **Servlets**

- ✓ Great for Java people
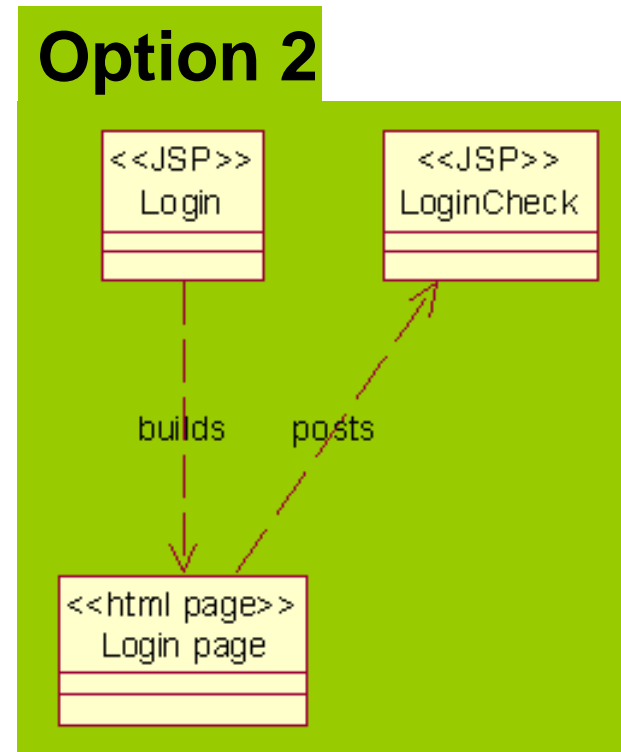- ✓ Difficult to manage graphical changes in HTML layout.

❖ **JSP**

- ✓ Great for web developers
- ✓ Seductive tendency to write logic in the JSP page.

JSP Model

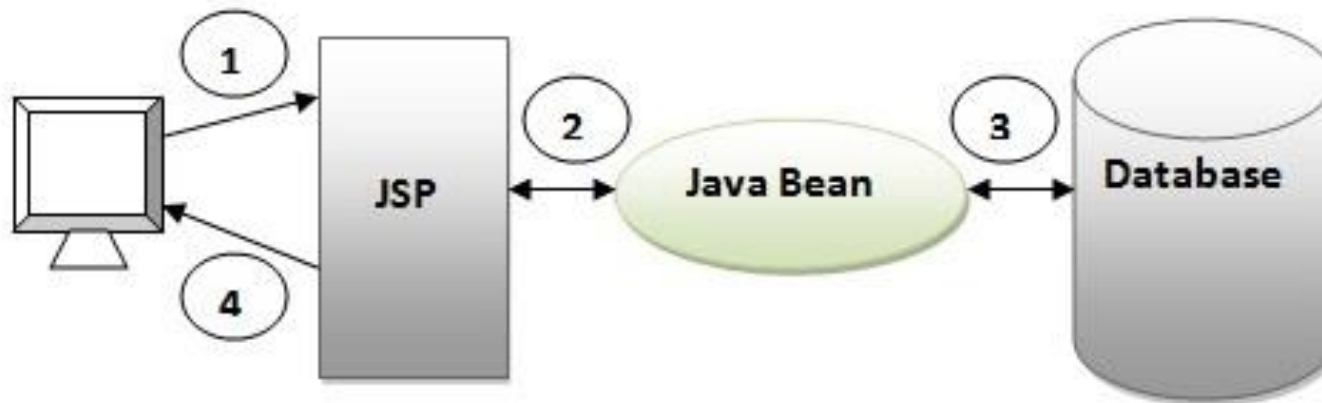❖ Web applications where JSP pages are used for every aspect of the development.

JSP Model 1

❖ A request is made to a JSP or servlet

❖ The JSP or servlet handles all responsibilities for the request
- ✓ processing,
- ✓ validating data,
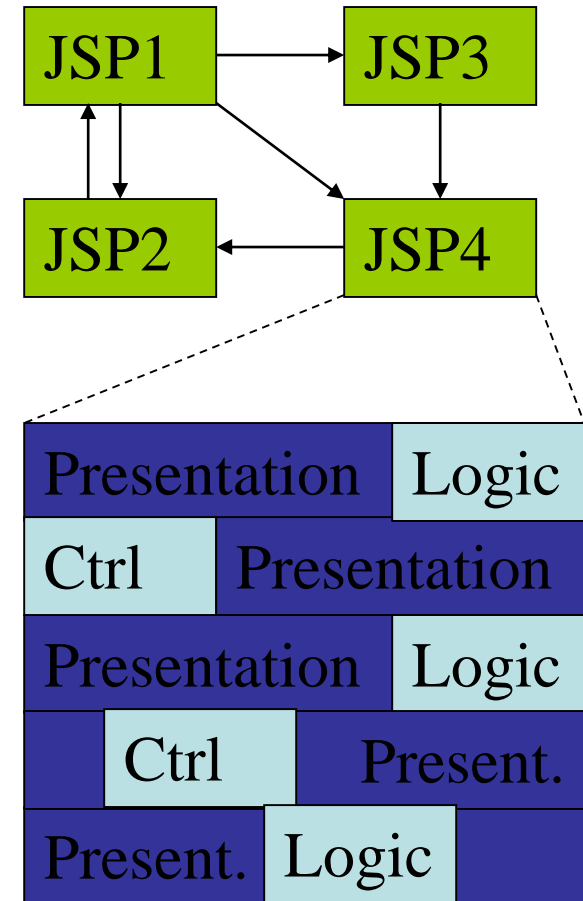- ✓ handling the business logic,
- ✓ and generating a response

JSP Model 1 Observation

❖ **The Good**

  ✓ Easiest Solution

❖ **The Bad**

  ✓ Presentation and Logic are mixed.

❖ **The Ugly**

  ✓ No reuse possibilities

JSP Model 1 Observation (cont.)
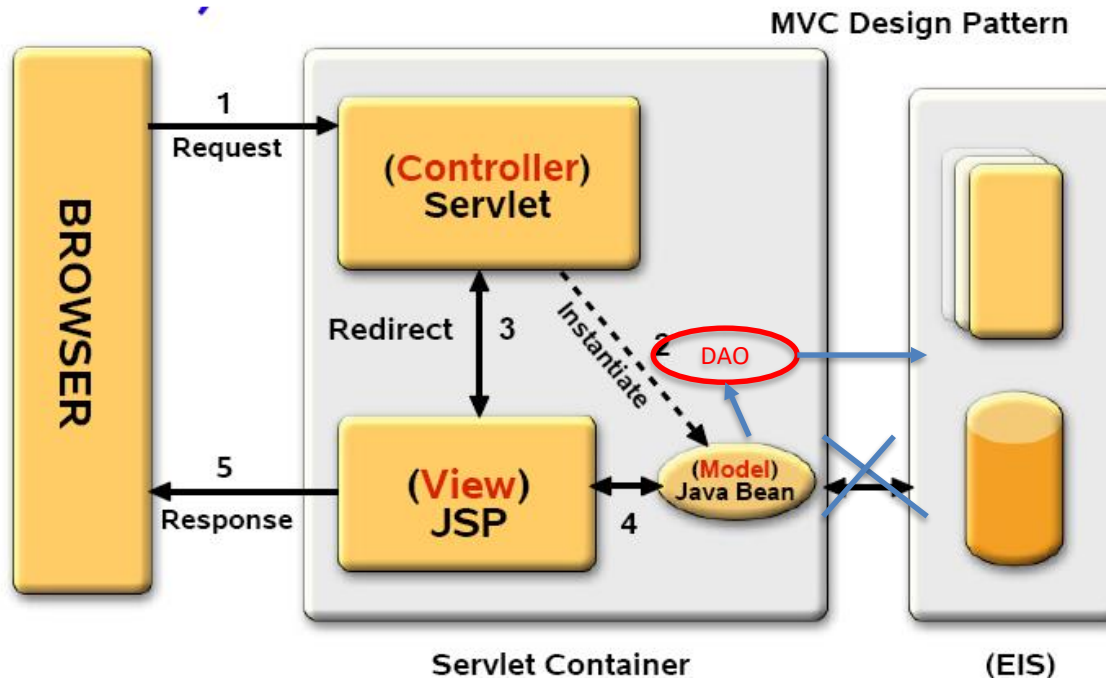
❖ **Advantages**

- ✓ Lightweight design – for small, static application
- ✓ Suitable for small applications having very simple page flow, little need for centralized security control and logging.

• **Limitations**

- ✓ Navigation Problem – to change name of JSP file have to change in many location
- ✓ Applications are difficult to modify – large Java code being embedded in JSP page
- ✓ Not suitable for large and complex applications

JSP Model 2

❖ Web applications where JSP pages are used for the GUI aspect of the web development

❖ The logic of the application is placed in the servlets it posts to.

JSP Model 2

❖ Model 2 separates the display of content from the logic used to obtain and manipulate the content.

❖ **Advantages**:

  ✓ Easier to build, maintain and extend: Suitable for large and complex applications.

  ✓ Single point of control (Servlet) for security and logging.

❖ **Limitations**: Increase Design Complexity

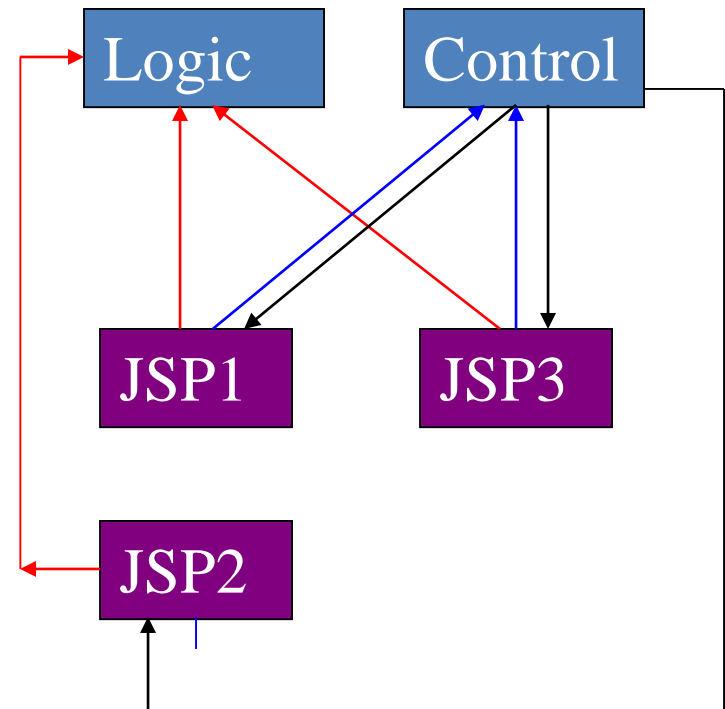# JSP Model 2 – Observation
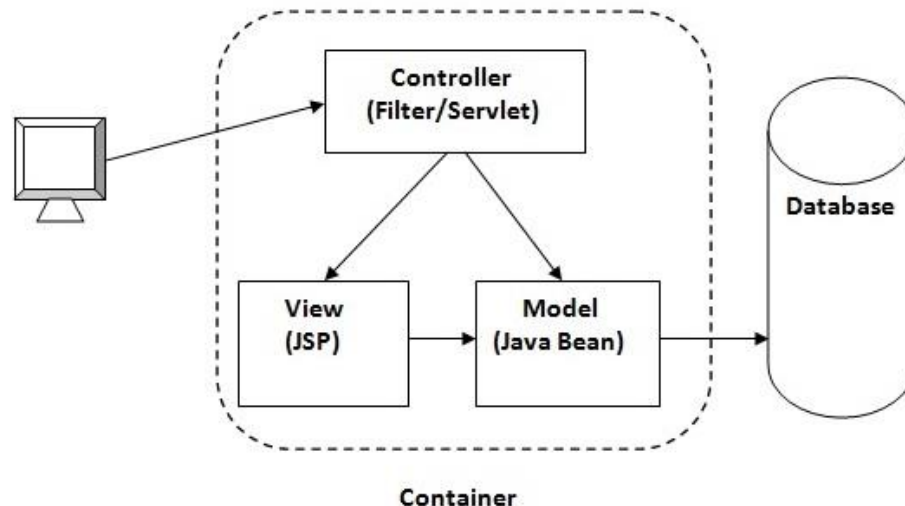
❖ **The Good**

  ✓ Reuse opportunities: Other application may be able to use the same code.

❖ **The Bad**

  ✓ There is no longer a one to one mapping from a view to a single source of code.

❖ **The Ugly**

  ✓ Takes more forethought.

Summary

⑩What Is a MVC?

⑩Web Application MVC Pattern

- ✓ Model
- ✓ View
- ✓ Controller

⑩JSP Model 1, 2

Thank you