# The University of Cambodia

## Homework Week 1 to Week 3

**Lecture Name:** TOUCH SARIM

**Course Name:** ITE402/ITE402K Spring Framework

**Student Name:** YOEURN SONITA

**Student ID:** 60-210370

### 1. What is Spring Framework?

The **Spring Framework** is a comprehensive, open-source framework for building Java applications. It simplifies enterprise-grade application development by providing:

- **Dependency Injection (DI)** for loose coupling.
- **Aspect-Oriented Programming (AOP)** for separating concerns.
- **Transaction management** and integration with other frameworks (e.g., **Hibernate**, **JPA, JDBC**).
- **Web development support** via **Spring MVC**.

### 2. Explaining the Project Structure in a Spring Project in Eclipse IDE:

In a Spring project, the common structure in **Eclipse IDE** is:

- **src/main/java/**: Contains application code, including **controllers**, **services**, **repositories**, and main configuration files.
- **src/main/resources/**: Holds non-Java files like **application.properties** for configuration, and static resources like HTML, CSS, and JS files.

- **src/test/java/**: Contains unit and integration tests, typically using **JUnit**.
- **pom.xml**: The Maven build configuration file that manages project dependencies and build settings.
- **target/**: Generated folder that contains compiled classes, packaged JARs/WARs, and other build outputs.

### 3. What is Spring Boot MVC?

**Spring Boot MVC** is a feature of **Spring Boot** that simplifies building web applications using the **Model-View-Controller (MVC)** pattern:

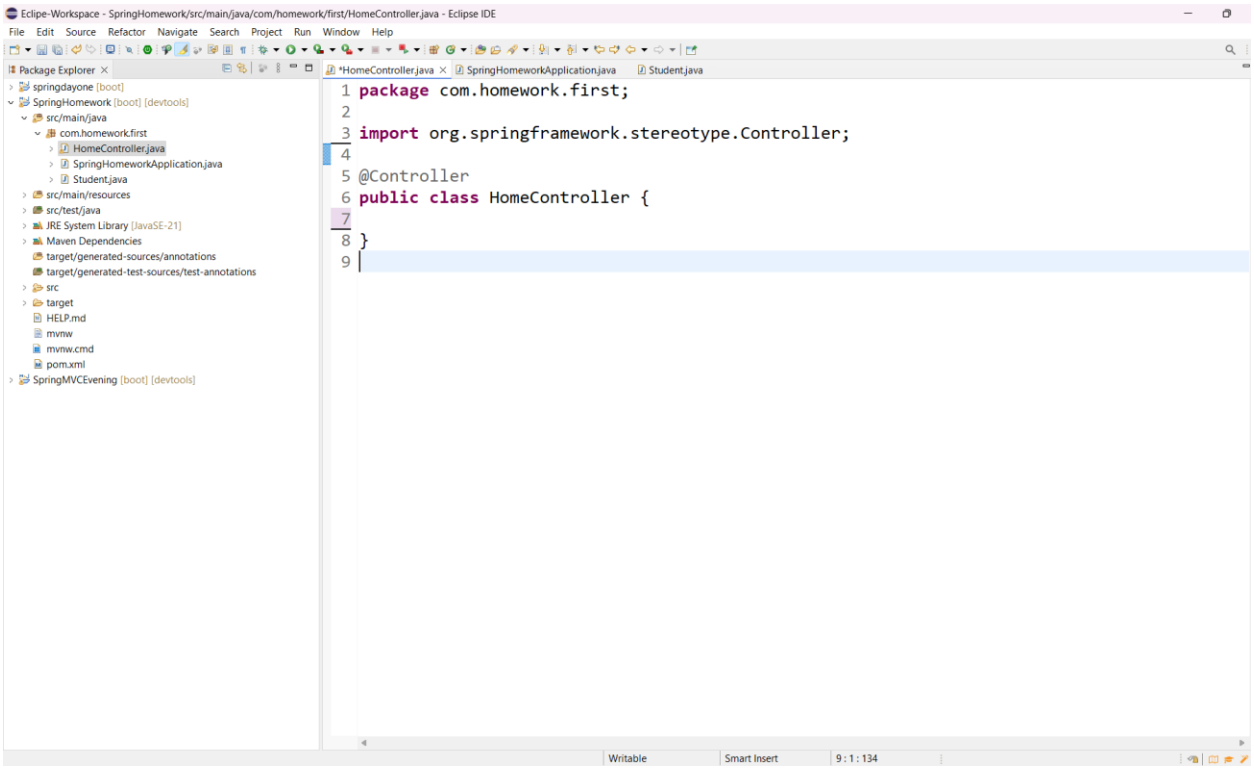
- **Model**: Represents application data and business logic.
- **View**: Displays the data to the user (e.g., using **Thymeleaf**).
- **Controller**: Handles HTTP requests, processes them, and returns a view or response. Spring Boot makes it easier to set up MVC with auto-configurations, embedded web servers, and integrated view technologies.

### 4. What are the Annotations in Spring MVC: @Controller, @GetMapping?

- **@Controller**:
    - Marks a class as a Spring MVC controller that handles HTTP requests and returns views.
- **@GetMapping**:
    - A specialized version of **@RequestMapping(method = RequestMethod.GET)**, it maps HTTP **GET** requests to specific handler methods in a controller.

## 5. Creating a controller class in spring boot project

SCREENSHOT:

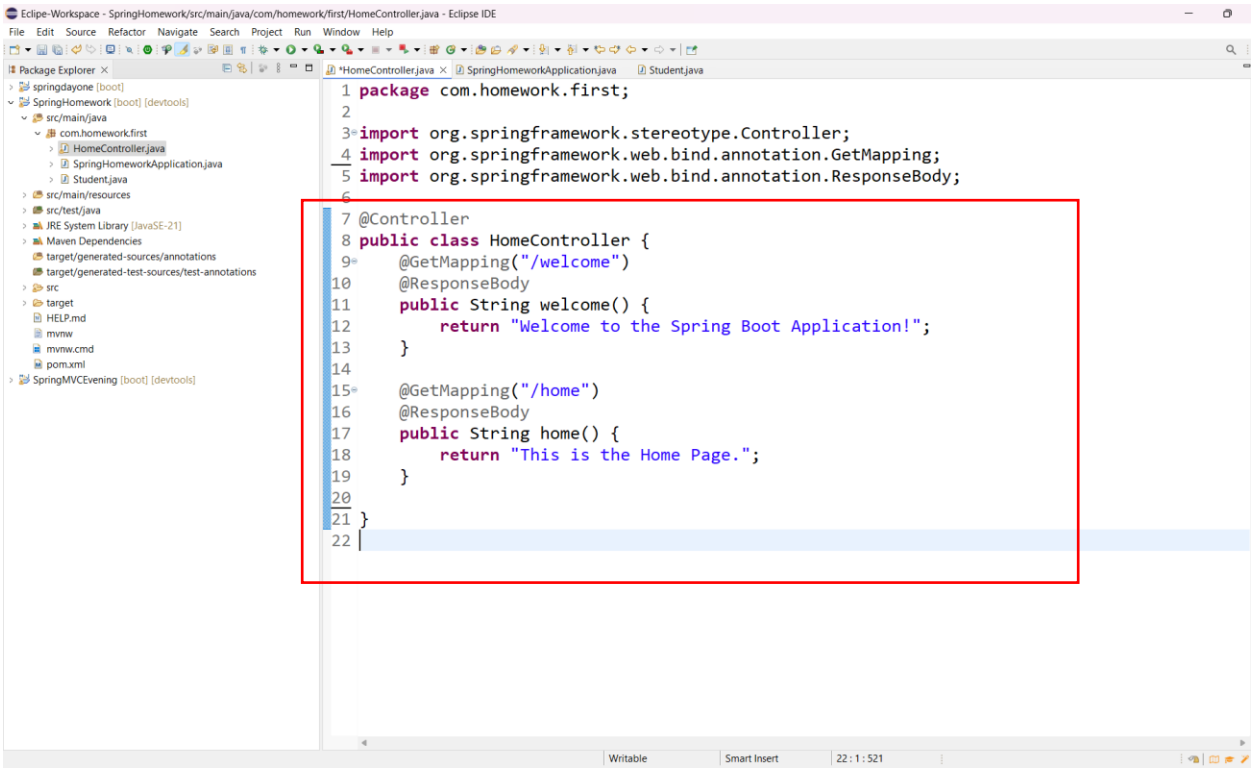



CODE:

```
package com.homework.first;

import org.springframework.stereotype.Controller;


@Controller

public class HomeController {


}
```

## 6. Create two methods to the URL (“/welcome”) and “/home”

SCREENSHOT:

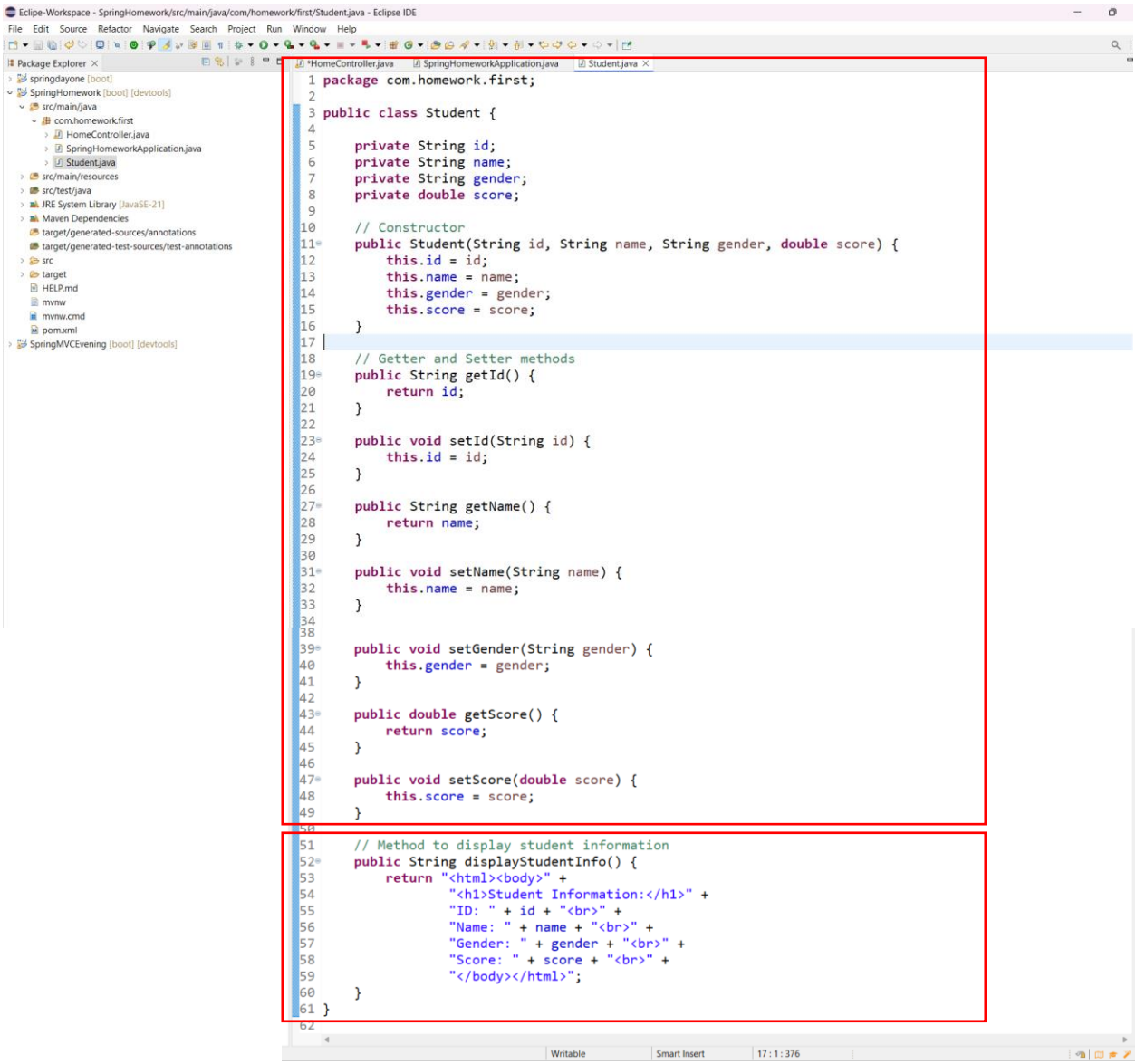


CODE:

```
package com.homework.first;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class HomeController {
      @GetMapping("/welcome")
    @ResponseBody
    public String welcome() {
        return "Welcome to the Spring Boot Application!";
    }

    @GetMapping("/home")
    @ResponseBody
    public String home() {
        return "This is the Home Page.";
    }
}
```

**7. Creating a java class student (id,name,gender, score) and setter and getter methos and a method to display all student information.**

SCREENSHOT:

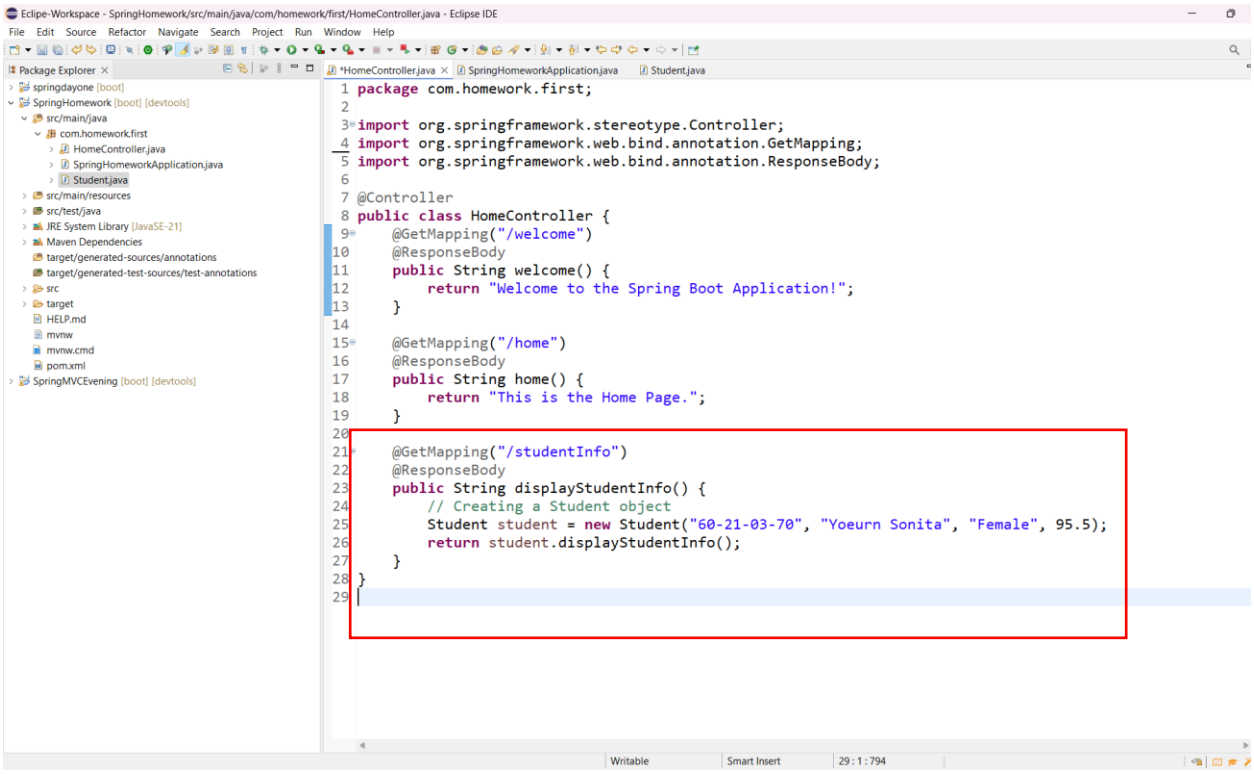


CODE:

```
package com.homework.first;


public class Student {


    private String id;
    private String name;
    private String gender;
    private double score;


    // Constructor
    public Student(String id, String name, String gender, double
score) {
        this.id = id;
        this.name = name;
        this.gender = gender;
        this.score = score;
    }
    // Getter and Setter methods
    public String getId() {
        return id;
    }


    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
}
```

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

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public double getScore() {
        return score;
    }

    public void setScore(double score) {
        this.score = score;
    }

    // Method to display student information
    public String displayStudentInfo() {
      return "<html><body>" +
                "<h1>Student Information:</h1>" +
                "ID: " + id + "<br>" +
                "Name: " + name + "<br>" +
                "Gender: " + gender + "<br>" +
                "Score: " + score + "<br>" +
                "</body></html>";
    }
}
```

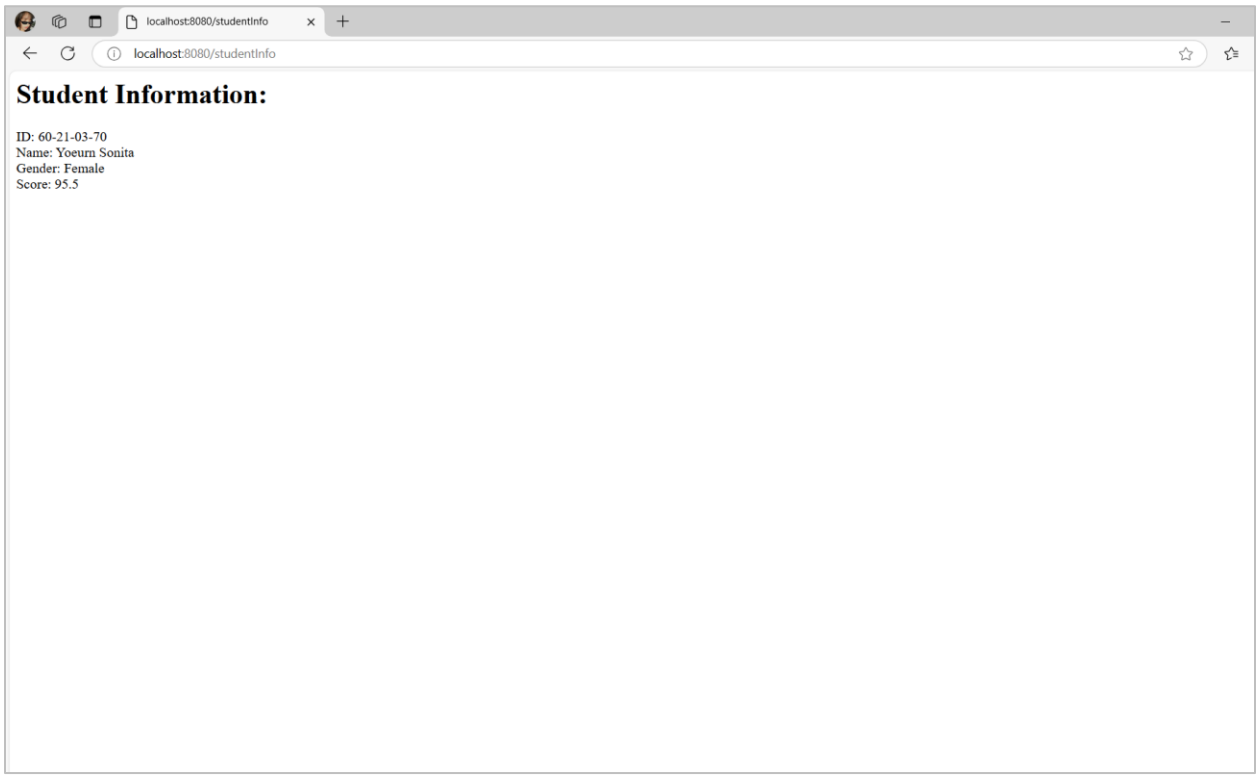
SCREENSHOT:


Eclipse-Workspace - SpringHomework/src/main/java/com/homework/first/HomeController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
springdayone [boot]
SpringHomework [boot] [devtools]
src/main/java
com.homework.first
HomeController.java
SpringHomeworkApplication.java
Student.java
src/main/resources
src/test/java
JRE System Library [JavaSE-21]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
SpringMVCEvening [boot] [devtools]
*HomeController.java
SpringHomeworkApplication.java
Student.java
1 package com.homework.first;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.ResponseBody;
6
7 @Controller
8 public class HomeController {
9 @GetMapping("/welcome")
10 @ResponseBody
11 public String welcome() {
12 return "Welcome to the Spring Boot Application!";
13 }
14
15 @GetMapping("/home")
16 @ResponseBody
17 public String home() {
18 return "This is the Home Page.";
19 }
20
21 @GetMapping("/studentInfo")
22 @ResponseBody
23 public String displayStudentInfo() {
24 // Creating a Student object
25 Student student = new Student("60-21-03-70", "Yoeurn Sonita", "Female", 95.5);
26 return student.displayStudentInfo();
27 }
28 }
29
Writable
Smart Insert
29 : 1 : 794

CODE:

```
package com.homework.first;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class HomeController {
      @GetMapping("/welcome")
    @ResponseBody
    public String welcome() {
        return "Welcome to the Spring Boot Application!";
    }

    @GetMapping("/home")
    @ResponseBody
    public String home() {
        return "This is the Home Page.";
    }

    @GetMapping("/studentInfo")
    @ResponseBody
    public String displayStudentInfo() {
        // Creating a Student object
        Student student = new Student("60-21-03-70", "Yoeurn
Sonita", "Female", 95.5);
        return student.displayStudentInfo();
    }
}
```

RESULT:


localhost:8080/studentInfo
localhost:8080/studentInfo
Student Information:
ID: 60-21-03-70
Name: Yoeurn Sonita
Gender: Female
Score: 95.5