**Student ID:** 610764

**Full Names:** Otgonbayar Mijiddorj

Web Application Programming

(CS472)

(April 2020)

Instructor: O. Kalu

Final Exam

1. The exam duration is 2 hours.
2. The exam is an online, computer-based exam; so you may use a computer for both the Part 1 (theory) and Part 2 (coding) tasks.
3. **This exam paper document is a copyrighted material and so it must not be copied or reproduced or transferred or shared or distributed**.
4. You are expected to use an IDE or any Code Editor tool of your choice to implement your solutions for the questions in the Part 2 (Web Application Coding).
5. During the exam, if you have any question, please **send it to me through a Chat** in the Microsoft Teams app.
6. Upon completion, put your entire Exam (including the projects/folders with your **source code** and this document with your typed-in answers) into a single zip file named **FinalExam.zip**, and submit to Sakai, under the Assignment titled, “Final Exam”.
7. **NOTE**: ***If you fail to submit your exam to Sakai because it has past the Submission due time, and you then email it instead, then be aware that your maximum possible score will be 83%,and only if your work scores up to that level.***

--------------------------------------------------------------------------------------------------------------------

Make sure to include the screenshots of your results, where it is required.

--------------------------------------------------------------------------------------------------------------------

(CS472 –WAP)

(April 2020)

Final Exam (40 points)

**Part I – Theory (True/False, Short answers, Multiple-choice questions):** (16 points)

**Note:** *For these questions, please follow the instructions given for each individual question and do type or mark all your answers right here in this document.*

1. (6 points) Answer the following questions with True or False. For each answer, give a rationale (i.e. If True state how, if False state why. No rationale, earns you just half of the points if your True/False answer is correct, and zero point if your True/False answer is incorrect).
   1. (2 point) HTTP is stateless; Hence Web Applications need additional mechanisms and techniques for Maintaining State.

True or False? **(Note: You must mention an example in your rationale)**

True

Rationale: HTTP protocol is stateless and every http request is new request to the server. In other words, request begins when client opened TCP socket to the server and ends when the server sends the last packet to the client and closes the connection. However, any web application recieves multiple HTTP request for particular user. That is why we need an additional mechanism to identify all these HTTP requests belong to that particular user.

* 1. (2 points) During execution of a web application that uses Java Servlet technology, a new instance of the Servlet class in created by the container, for every HttpRequest received.

True or False?

False

Rationale: Container creates new instance of servlet when container receives the new request for a servlet. And container obtains a new thread and calls service method on HttpServlet instance in thread. In other words, shared one instance with multiple threads.

* 1. (2 points) In Java web application programming, the statements,

*RequestDispatcher reqDispatcher = request.getRequestDispatcher(“stuRegForm.jsp”);*

*reqDispatcher.forward(request, response);*

causes the web browser to resend a new HttpRequest to the web page named, stuRegForm.jsp.

True or False?

False

Rationale: Browser is completely unaware of servlet forward. Hence browser will not resent a new HTTP request to the web page to stuRegForm.jsp

1. (8 points) Give short answers (and citing examples where required) to the following questions.
   1. (2 points) What is the difference between a Web Server and a Web Container? Give an example for each.

Answer: Web server serves static content such as images and responses cannot be customized or modified based on input data from the client.

Web container contains servlets, which are java code for our case and that can handle http requests and return dynamic content.

Web server examples are Apache, NGINX

Web container examples are tomcat, glassfish, JBoss..

* 1. (2 points) Given in the table below, is the data for Employees to be stored in an HRM web system.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Employees** | | | | | | | | |
| EmpId | SSN | FullName | PhoneNo | DateOfBirth | DateOfEmployment | Level | Salary | Email |
| 1 | 123-45-67 | Usain Bolt |  | 1996-12-3 | 2014-1-4 | 16 | $15,005.95 |  |
| 2 | 123-45-68 | Anna-Marie Rodriguez | (641) 451-0001 | 2000-5-21 | 2014-11-9 | 13 | $4,562.00 | [amrod@gmail.mx](mailto:amrod@gmail.mx) |
| 3 | 123-45-69 | Victor Luiz Garcia |  | 1998-11-7 | 2013-11-7 | 12 | $17,750.05 |  |

Express and write-out, in the space below, the Employees data in JSON format.

[{"EmpId":1,"SSN":"123-45-67","FullName":"Usain Bolt","PhoneNo":"","DateOfBirth":"1996-12-3","DateOfEmployment":"2014-1-4","Level":16,"Salary":"$15,005.95","Email":""},

{"EmpId":2,"SSN":"123-45-68","FullName":"Anna-Marie Rodriguez","PhoneNo":"(641) 451-0001","DateOfBirth":"2000-5-21","DateOfEmployment":"2014-11-9","Level":13,"Salary":"$4,562.00","Email":"amrod@gmail.mx"},

{"EmpId":3,"SSN":"123-45-69","FullName":"Victor Luiz Garcia","PhoneNo":"","DateOfBirth":"1998-11-7","DateOfEmployment":"2013-11-7","Level":12,"Salary":"$17,750.05","Email":""}]

* 1. (1 point) How is a Java Server Page related to Servlet?

JSP can contain both of HTML code and Java code. JSP pages are compiled into part of Java servlet when container runs.

* 1. (3 points) Name and briefly explain 3 mechanisms for maintaining state in Web applications.

1. **Request scope:** short term computed results to pass from one servlet to another. For example, forward request to another servlet. Life time is for a request-response cycle

2. **Session scope:** Conversational state info across a series of sequential requests from a particular user. All info in this session is available only for this particular user and container manage this info to identify this user and use to define actions for this particular user. Lifetime is throughout the session.

3. **Application/context scope:** global info available to any other users or servlet in this application. This information is available for all users of the servlet and web application. Lifetime is for the lifetime of the application.

1. (2 points) The following questions involve multiple choices; choose the correct option(s) by marking with green highlight/background.
   1. (1 point) For a JSP custom tag whose Tag Library Descriptor (TLD) specifies the following:

<tag>

**<name>showCardBox</name>**

**<tag-class>edu.miu.test.cardBoxTag</tag-class>**

**<body-content>*scriptless*</body-content>**

</tag>

Which is a correct usage of the tag on a JSP page?

**Option A**.

<cbx:showCardBox><%= request.getAttribute(“msgBox”) %></cbx:showCardBox>

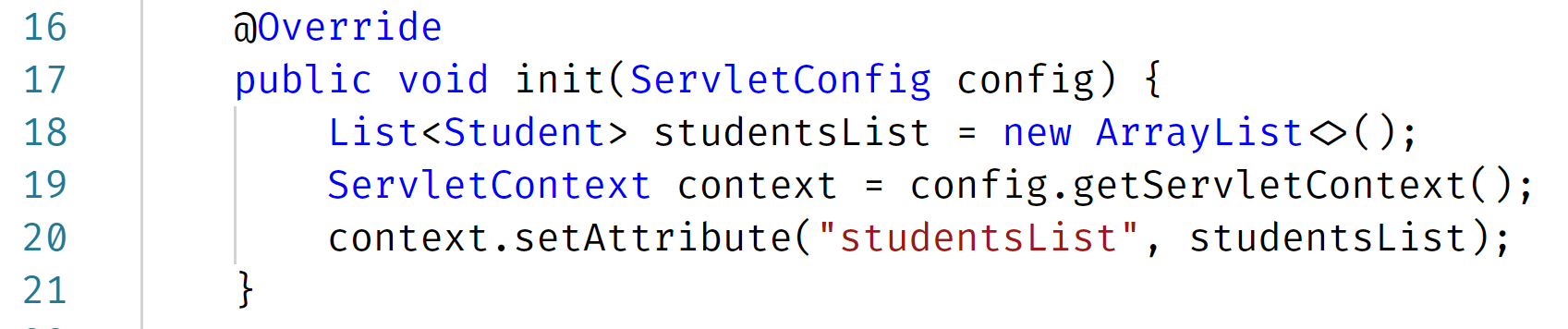
**Option B**. Rationale: body-content is scriptless

<cbx:showCardBox>${msgBox}</cbx:showCardBox>

**Option C**.

<cbx:showCardBox><% request.getAttribute(“msgBox”) %></cbx:showCardBox>

* 1. (1 point) Consider the following Java Servlet code fragment:

D

When the code is executed, the attribute named, ‘studentsList’, will be available on which scope?

A. Session scope

B. Request scope

C: Application scope - Rationale: studentsList will be created only once when servlet receives new request and available to all users. So, application scope.

D. Page scope

**Part II – Web Application Coding Skills:** (24 points)

**Note:** *For the tasks in this question, where applicable, you are expected to take screenshot(s) of your web UI(s), save into a .png or .jpg image file, placed inside a folder named, screenshots and include these in the FinalExam.zip file, you submit or copy and paste your screenshots to the bottom of the associated question(s) right here in this document.*

1. (24 points) The figure marked as Figure x displayed below, is of a Student Registration web form. The content of the drop-down list data field (labeled, Next Course to take) is shown in Figure y, further below.

atio

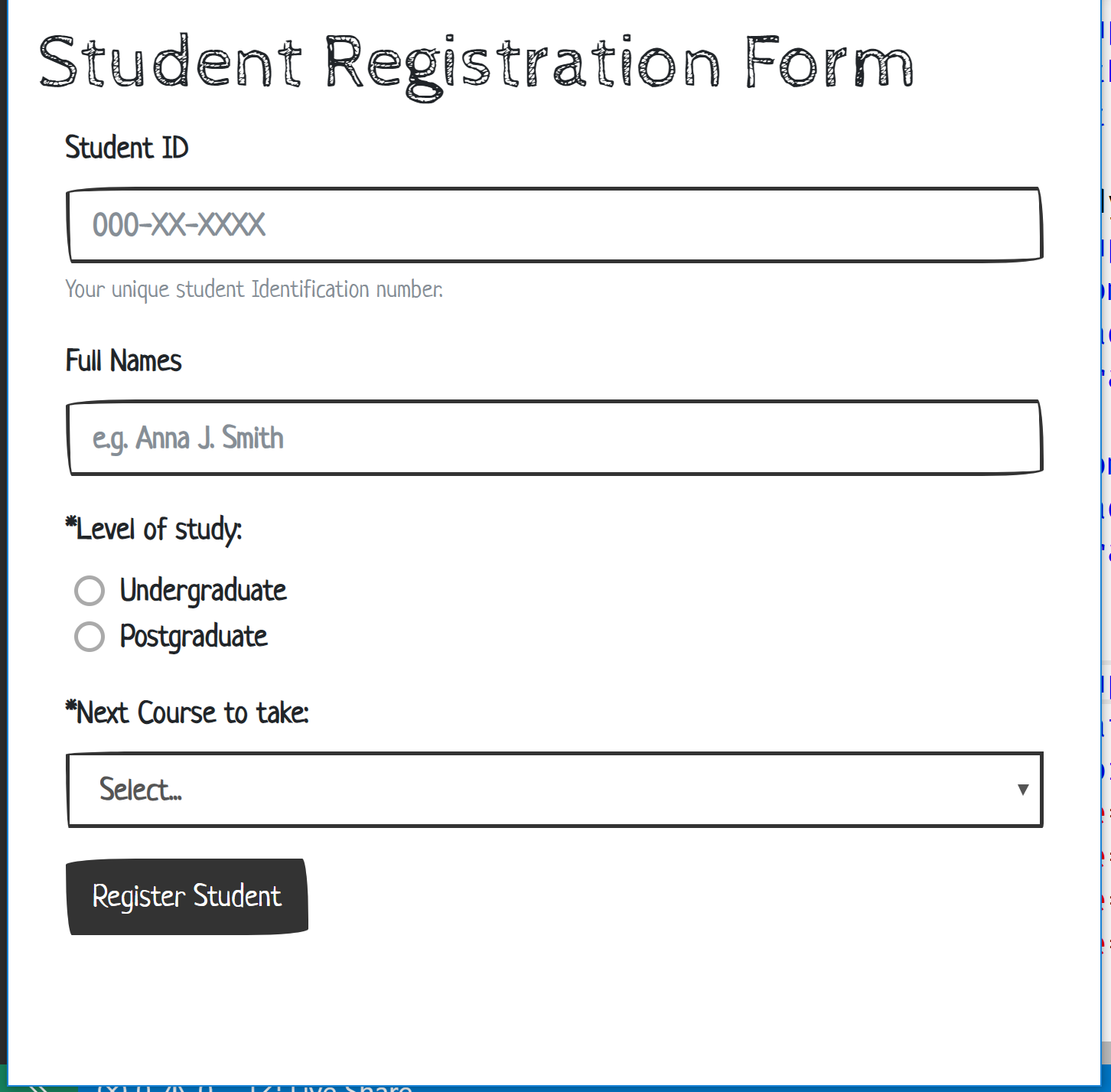


Figure x

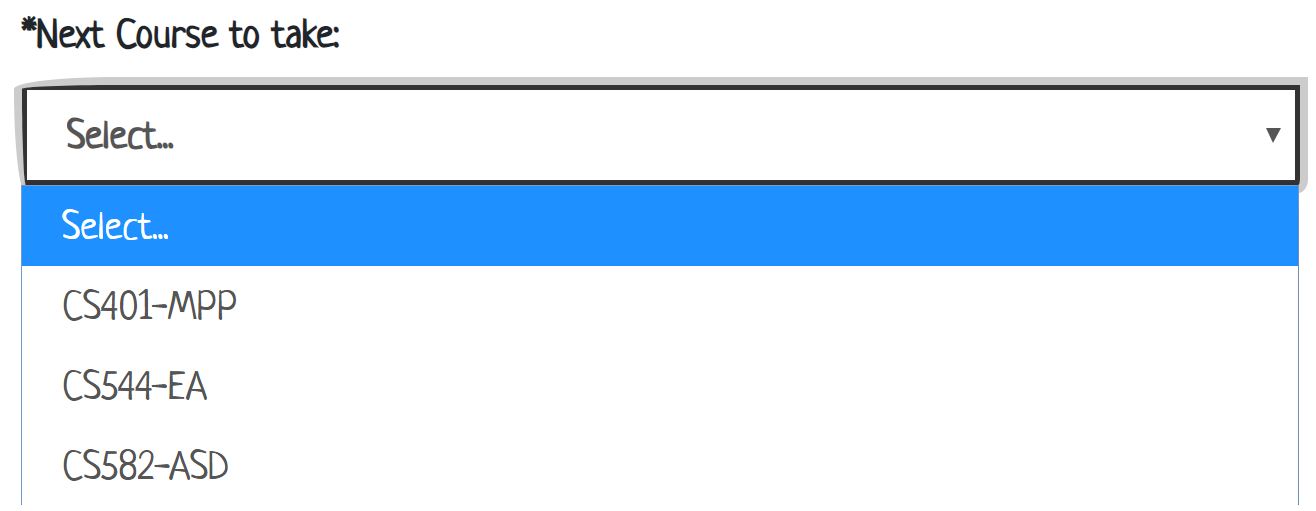


Figure y

Now, build a Web Application that presents a user (e.g. the Registrar) with the above web form and provides the following features and functionalities:

1. Using HTML5 (and optionally, CSS), code the appropriate markup for the form, as shown in Figures x and y above. Note: Your UI does not necessarily have to be exactly the same as the sample shown, but it should have all the necessary form data input fields, labels etc.
2. Student ID, Full Name, Level of study and Next Course are all required data needed and mandatory for registering a Student into the system.
3. Add validation using appropriate regular expression to ensure that any Student ID entered must be in the specified format of, 000-XX-XXXX, as shown on the form UI above. And where X is a numeric digit.
4. Upon entering all the required data in the web form, when the user clicks the “Register Student” button, the form data should be submitted/transmitted to a backend/server-side code implementation which does the following:
   1. Stores all the Registered Student data in memory. Note: You are expected to choose/use an appropriate data structure and implement an appropriate in-memory data storage mechanism.
   2. Prints-out each new Student Registration data to the console on the server.
5. Upon submission of the Student Registration form data, the form data fields should be cleared.
6. The backend/server-side code should be implemented using the Model 2 (MVC) architecture.

//-- The End --//