**Student ID:** 985892

**Full Names:** Dulamkhand Batjargal

Web Application Programming

(CS472)

(September 2017)

Instructor: Obinna A. Kalu

W3D6 – Exam 3

1. The exam duration is 2 hours.
2. The exam is a computer-based exam.
3. You are expected to use a CS lab or your own computer to answer both the Coding questions and the theory/non-coding/knowledge-based questions. You may use the Internet and/or the lecture slides for reference purposes to lookup APIs or code syntax.
4. Make sure to switch-off your cell-phones or simply turn the ringer off.
5. You may use blank sheet(s) of paper for your scratch work, if needed.
6. Exams are copyrighted materials and must not be copied, reproduced or redistributed.
7. All answers to the theory/non-coding/knowledge-based questions should be typed-in, on this document, following the questions.
8. All answers to the Coding questions may be typed-in as source code files, using a Code Editor or IDE. But be sure to copy your finished code for each coding question from your Code Editor and paste it to this document as your answer.
9. Finally, compress/zip your entire code folder into one zip file and upload/submit it to Sakai, along with your typed/pasted answers in this document.

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

(CS472 - WAP)

(September 2017)

W3D6 – Examination 3

**Part I – Science of Consciousness (SCI):** (4 points)

Given in the table below, are some principles from the Science of Consciousness (SCI), related topics in Web Application Programming (WAP) and Connection sentence(s). For each section, fill-in the blank cell(s), by either providing the missing principle from the Science of Consciousness or a Web Programming topic or a connection sentence.

**Note**: To get the full credit, make sure your connection sentence provides a reasonable, clear link/relation between the stated SCI principle and the WAP topic.

|  |  |  |
| --- | --- | --- |
| **SCI principle** | **Topic from Web Application Programming (WAP)** | **Connection sentence(s)** |
| Complete knowledge is the wholeness of the Knower, the Known and the Process of Knowing. | A Web Application based on the Model 2 design is one that implements the Model-View-Controller (MVC) architecture. | MVC architecture combines major components allowing for efficient code reuse and parallel developments which represents whole complete developments of the system. |
| Purification leads to progress. | Security – DAO Prepared Statements. | Always use prepared statements, separate your concerns. |
| At higher states of consciousness, actions become more powerful | HTTP is stateless; Hence Web Applications need additional mechanisms for Maintaining State. | HTTP is a stateless protocol (an HTTP request by its own nature does not know anything about previous/other requests). Web applications must therefore explicitly manage state information from one request to the next (like “is this person logged in”, or even “who is this person”). |

**Part II – Theory (Short answers, True/False, Multiple-choice questions):** (36 points)

1. (8 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 points) In Web Application security, Data Access layer code written with the JDBC API and using Prepared Statements can be susceptible to SQL Injection attack.

False - Prepared Statements correctly uses parameterized queries; binding variables and corresponding setString methods, SQL Injection can be easily prevented.

* 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.

False - The container calls service() method to process a client's request in new thread for every request. But only one instance of Servlet class will be created.

* 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.

False – Forward does not create new HttpRequest, but resends the old request to the jsp page.

* 1. (2 points) Every JSP page contains an implicit object named, out, which represents a reference to the output PrintWriter.

True – JSPs have implicit object or pre-defined variable called ‘out’/PrintWriter which send output to the client.

1. (24 points) Give short answers (and examples where required) to the following questions.
   1. (3 points) What is the difference between a Web Server and a Web Container?

Web Container or Servlet Container or Servlet Engine: is used to manage the components like servlets, JSP. It is a part of the web server. It is used for dynamically generate the web pages on the server side.

Web Server or HTTP Server: a server which is capable of handling HTTP request send by a client and respond back with a HTTP response.

* 1. (3 points) What is the difference between the statements: response.sendRedirect(“contactForm.jsp”) versus requestDispatcher.forward(“contactForm.jsp”).

First line is redirecting while second is forwarding. Forward passes the request to another resource on the server. Redirect - server sends http status code 3xx to client along with the redirect URL usually 302 temporary redirect.

* 1. (9 points) With regards to Java Web application programming:
     1. (3 points) What is a Servlet? Give an example with code snippet.

A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model.

Example:

@WebServlet(description = "AccFormServlet", urlPatterns = {"/form"})

public class AccFormServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public AccFormServlet() {

super();

}

@Override

public void init() throws ServletException {

super.init();

}

}

* + 1. (3 points) What is a Java Server Page? Give an example with code snippet.

Java Server Page is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications.

Example:

<%@ page language="java" contentType="text/html;charset=UTF-8" %>

<%@ page import="java.util.Date"%>

<html>

<head>

<title>Document</title>

</head>

<body>

Current date is <%= new Date() %>

</body>

</html>

* + 1. (3 points) What is the relationship between a Servlet and a Java Server Page?

JSP pages actually are compiled into Java servlets. You don't compile JSPs, your Java-enabled server performs the compilation from JSP page into a Java servlet for you.

* 1. (3 points) In Java Web Application programming, how do HttpRequests relate to the doGet and doPost methods in a Servlet? (You may give an example to illustrate your answer).

HTTP requests send input parameters as name/value pairs. Input parameters are text that must be accessed and converted by a servlet. This is the main mechanism web apps use to send information from the browser to the server.

* 1. (6 points) Name and explain 3 techniques for maintaining state in Web applications.

Session - Servlets provide a convenient and stable session-tracking solution using the HttpSession API (remember we discussed about HttpSession Object.

Cookies -When a client browser makes a request for a web page from a server, it first checks to see if there are any cookies present from that server in its cookie folder. If there are any cookies then those cookies are sent to the web server along with the request for the page. The application program on the server side reads the cookies and tailors the web page accordingly. It may reset the cookies according to the present transaction.

Hidden fields are name-value pairs that are embedded in a form that can be checked by the script processing form data on the server side. As the name implies hidden variables are not displayed when the form is rendered on the client side.

1. (4 points) The following questions involve multiple choices; choose the correct option by putting a green highlight over, either Option A or Option B or Option C or Option D.
   1. (2 points) Consider the following Java servlet code snippet:



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

**Option A**.

Session scope

**Option B**.

Request scope

**Option C**.

Application scope

**Option D**.

Page scope

* 1. (2 points) For a JSP custom tag whose Tag Library Descriptor (TLD) specifies the following:

<tag>

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

**<tag-class>edu.mum.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**.

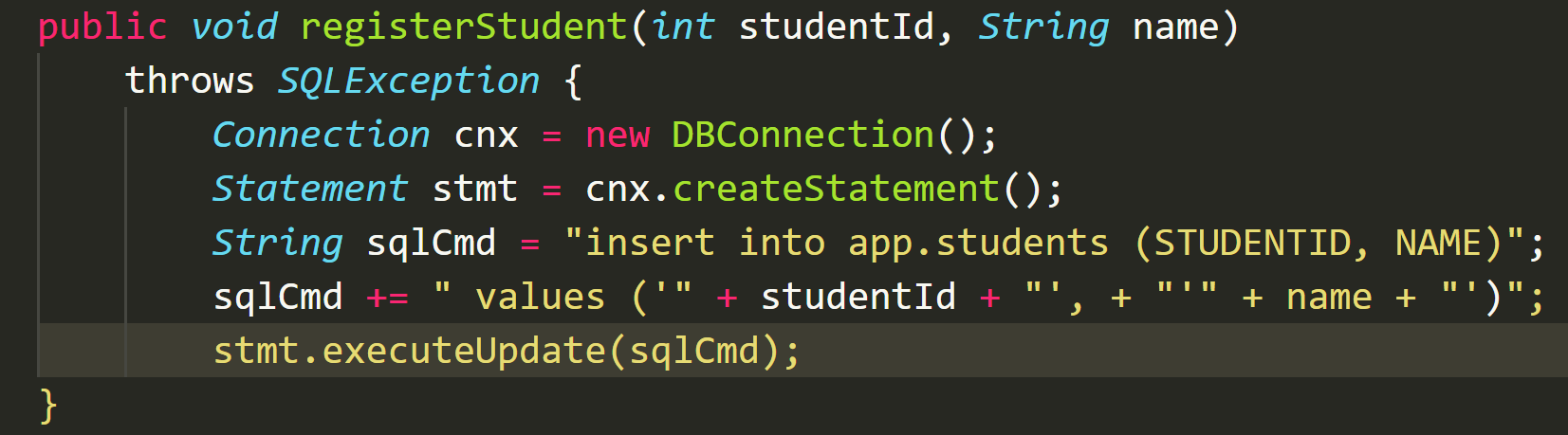
<cbx:showCardBox>${msgBox}</cbx:showCardBox>

**Option C**.

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

**Part III – Skill (JavaScript/Web Coding):** (60 points)

1. (10 points) Re-write the JDBC code snippet given below, by applying a technique that prevents an SQL Injection attack. i.e. Re-implement the method named, registerStudent, and make necessary changes to the JDBC code, to make it more secure and less vulnerable to SQL injection.



Answer:

public void registerStudent(int studentId, String name) throws SQLException {

Connection cnx = new DBConnection();

String sql = "INSERT INTO studeapp.studentsnts VALUES(NULL, ?, ?)";

try (PreparedStatement ps = cnx.prepareStatement(sql)) {

ps.setInt(1, studentId);

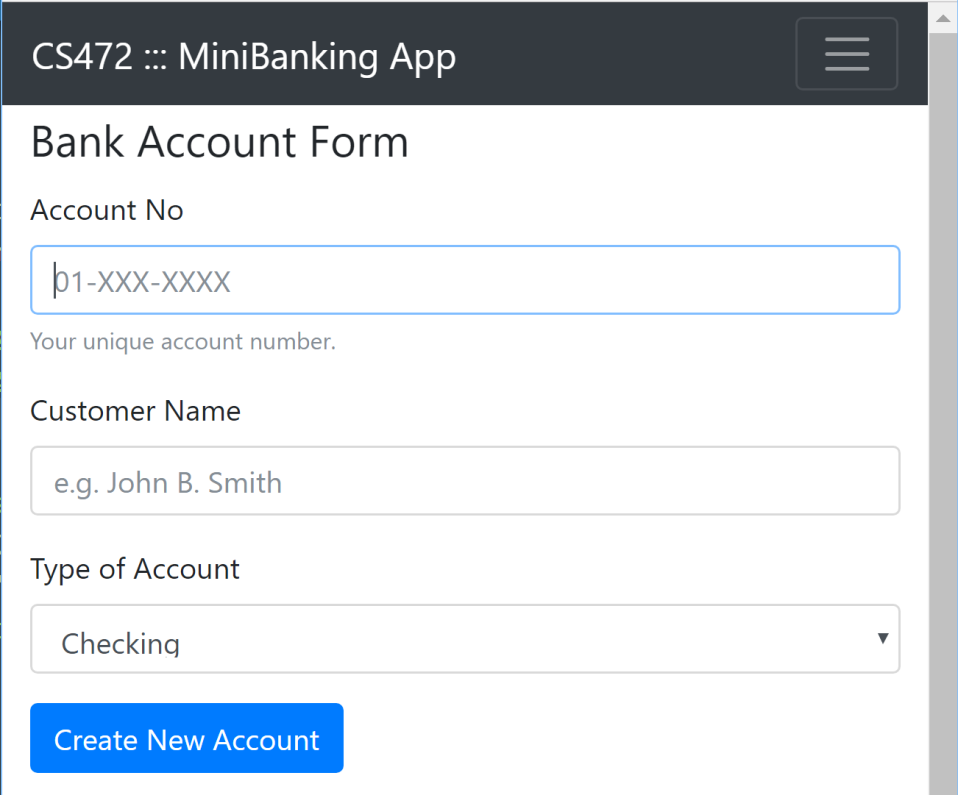
ps.setString(2, name);

ps.executeUpdate();

}

}

1. (20 points) Create a Java web application project and in it implement code for a Java Servlet named, AccountForm, that when invoked with the url, <http://hostname:portNum/WebApp/accountForm>, from the browser, will display the following web page:



*Note: hostname, portNum and WebApp will be as it applies to your solution.*

**Project folder: P3Q2**

<http://cs_labv47a-203:8080/P3Q2/accountForm>

**http://localhost:8080/WEB-INF/**

1. (30 points) Re-implement the above project by replacing the AccountForm servlet with a Java Server Page named, accountForm.jsp, which presents/displays the same Bank Account form as shown above. In this new webapp project, include the following:

a. A servlet named, ProcessAccountFormData, to which the accountForm JSP page submits its form data.

b. A Java Server page named, thankyou.jsp, that displays a thank you message and the Account data that was submitted.

**Note: There should be no Java code scriptlets in your JSP web pages code. Use only JSTL tags and Expression Language (EL) expressions.**

**Project folder: P3Q3**

**http://cs\_labv47a-203:8080/P3Q3/accountForm**

**http://localhost:8080/P3Q3/**

**//-- The End --//**