Student ID_____Student Name_____

Web Applications Architecture and Frameworks DE

Midterm Exam January 9, 2016

PRIVATE AND CONFIDENTIAL

1. **Allotted exam duration is 2 hours.**
2. **Closed book/notes.**
3. **No personal items including electronic devices (cell phones, computers, calculators, PDAs).**
4. **Cell phones must be turned in to your proctor before beginning exam.**
5. **No additional papers are allowed. Sufficient blank paper is included in the exam packet.**
6. **Exams are copyrighted and may not be copied or transferred.**
7. **Restroom and other personal breaks are not permitted.**
8. **Total exam including questions and scratch paper must be returned to the proctor.**

6 blank pages are provided for writing the solutions and/or scratch paper. All 6 pages must be handed in with the exam

BE VERY CAREFUL WITH THE GIVEN 2 HOURS AND USE YOUR TIME WISELY. THE ALLOTED TIME IS GIVEN FOR EVERY QUESTION.

Write your name and student id at the top of this page.

Question 1: [5 points] {10 minutes}
Give the 3 different scope types that you can use with servlets, and explain how they differ from each other.

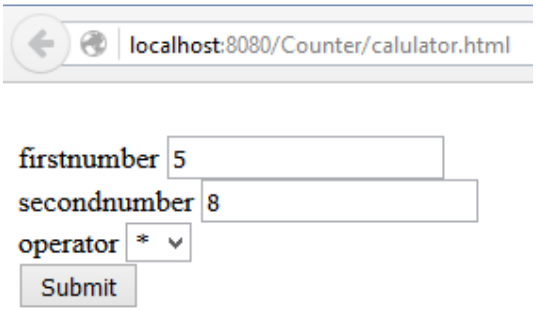
Request

Session

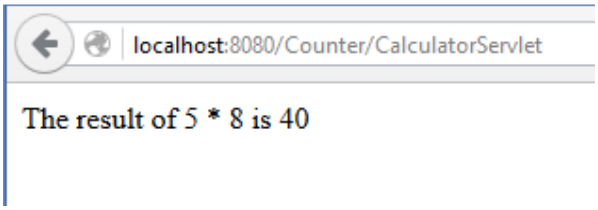
Application

Question 2: [15 points] {20 minutes}
Suppose we have the following form:

```
<form method="post" action="CalculatorServlet">
  <br>
  firstnumber  <INPUT TYPE="TEXT" NAME="firstnumber">
  <br />
  secondnumber  <INPUT TYPE="TEXT" NAME="secondnumber">
  <br />
  operator
  <select name="operator">
   <option label="+" value="+">+</option>
   <option label="-" value="-">-</option>
   <option label="*" value="*">*</option>
   <option label="/" value="/">/</option>
  </select>
  <br/>
  <input type="submit">
</form>
```



If you fill in 2 numbers, select an operator and then click the submit button, you should see the following result.



Write the CalculatorServlet so that this servlet shows the webpage given above. For this question, you are only allowed to write a servlet, you are **NOT** allowed to use JSP or JSF.

```
public class CalculatorServlet extends HttpServlet {
  protected void doPost(HttpServletRequest request, HttpServletResponse
                        response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String strFirstnumber = request.getParameter("firstnumber");
    String strSecondnumber = request.getParameter("secondnumber");
    String operator = request.getParameter("operator");
    int firstnumber = Integer.parseInt(strFirstnumber);
    int secondnumber = Integer.parseInt(strSecondnumber);
    int result = 0;
    if (operator.equals("+")){ result=firstnumber+secondnumber;}
    if (operator.equals("-")){ result=firstnumber-secondnumber;}
    if (operator.equals("*")){ result=firstnumber*secondnumber;}
    if (operator.equals("/")){ result=firstnumber/secondnumber;}
    out.println("<html>");
    out.println("<head><title>calculatorservlet</title></head>");
    out.println("<body>");
    out.println("The result of " +firstnumber+ " "+operator+" "+secondnumber+"
              is "+result);
    out.println("</body>");
    out.println("</html>");
  }
}
```

Question 3: [10 points] {10 minutes}

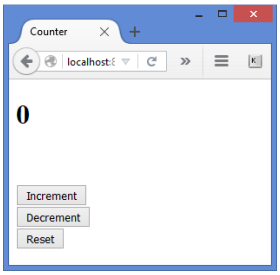
Suppose we have an ArrayList of Student objects, and a Student class has the attributes **firstName**, **lastName** and **dateOfBirth**. In the Servlet we place this ArrayList of Students in the request object and then we call a **jsp** file to display this list of students.

Write the code snippet that shows the list of students on the webpage. It is **NOT allowed to write JAVA code** in the JSP page (use JSTL and JSP EL).

```
<c:forEach var="student" items="${students}">
    ${student.firstName}
    ${student.lastName}
    ${student.dateOfBirth}
     <br/>
  </c:forEach>
```

Question 4 [35 points] {40 minutes}

Write a simple counter application using Servlets and JSP's according the MVC structure. When you start the application you see the following page:



The counter works as follows:
When you click the Increment button, the counter value is incremented according the following algorithm:

- If the value of the counter is smaller or equal than 5, the counter is incremented with 1.
- If the value of the counter is larger than 5 but smaller or equal than 10, the counter is incremented with 2.
- If the value of the counter is larger than 10, the counter is incremented with 3.

When you click the Decrement button, the counter value is decremented according the following algorithm:

- If the value of the counter is smaller or equal than 5, the counter is decremented with 1.
- If the value of the counter is larger than 5 but smaller or equal than 10, the counter is decremented with 2.
- If the value of the counter is larger than 10, the counter is decremented with 3.

When you click the Reset button, the counter value is set to 0.

Your implementation should follow the following requirements:

1. The application should follow the correct **Model-View-Controller** principles using Servlets, JSP's and Java classes.
2. You are not allowed to use hidden form fields
3. You are **not** allowed to use JSF.
4. It is **not** allowed to write JAVA code in the JSP page (use JSTL and JSP EL).
5. **For JSP pages, you only need to write the code between the <body> and </body> tags. Do not write the code for namespaces**

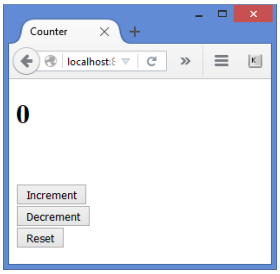
```
public class Counter {
   private int value=0;

   public void increment() {
      if (value <= 5) {
         value = value + 1;
      } else {
         if (value > 5 && value <= 10) {
            value = value + 2;
         } else {
            if (value > 10) {
               value = value + 3;
            }
         }
      }
   }
   public void decrement(){
      if (value <= 5) {
         value = value - 1;
      } else {
         if (value > 5 && value <= 10) {
            value = value - 2;
         } else {
            if (value > 10) {
               value = value - 3;
            }
         }
      }
   }
   public void reset(){
      value=0;
   }

   public int getValue() {
      return value;
   }

}
```

```
@WebServlet(urlPatterns = {"/IncrementServlet"})
public class IncrementServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      HttpSession session = request.getSession();
      Counter counter = (Counter) session.getAttribute("counter");
      if (counter == null){
         counter = new Counter();
         session.setAttribute("counter", counter);
      }
      counter.increment();
      RequestDispatcher view =request.getRequestDispatcher("counter.jsp");
       view.forward(request, response);
   }
}

@WebServlet(urlPatterns = {"/DecrementServlet"})
public class DecrementServlet extends HttpServlet {

   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      HttpSession session = request.getSession();
      Counter counter = (Counter) session.getAttribute("counter");
      if (counter == null){
         counter = new Counter();
         session.setAttribute("counter", counter);
      }
      counter.decrement();
      RequestDispatcher view =request.getRequestDispatcher("counter.jsp");
       view.forward(request, response);
   }
}

@WebServlet(urlPatterns = {"/ResetServlet"})
public class ResetServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      HttpSession session = request.getSession();
      Counter counter = (Counter) session.getAttribute("counter");
      if (counter == null){
         counter = new Counter();
         session.setAttribute("counter", counter);
      }
      counter.reset();
      RequestDispatcher view =request.getRequestDispatcher("counter.jsp");
       view.forward(request, response);
   }
}
```

Counter.jsp

```
<body>

    <h1>${counter.value}</h1><br/><br/>
    <form method='get' action='IncrementServlet'>
      <input type='submit' value='Increment' />
    </form>
    <form method='get' action='DecrementServlet'>
      <input type='submit' value='Decrement' />
    </form>
    <form method='get' action='ResetServlet'>
      <input type='submit' value='Reset' />
    </form>
  </body>
</html>
```

Question 5. [30 points] {30 minutes}
Write the counter application of the previous question using JSF.



Your implementation should follow the following requirements:

1. The application should follow the correct **Model-View-Controller** principles using JSF.
2. You are **not** allowed to use HTML, JSP's or servlets.
3. For this JSF application we will use annotation based configuration, so there is no faces-config.xml file. **Add the necessary annotations to the code**.
4. **Do NOT write getter and setter methods!**
5. **For web pages, you only need to write the code between the <body> and </body> tags. Do not write the code for namespaces**

```
    <h:body>
        <h:form>
        <h:outputText value="#{counterBean.counter.value}"/><br/><br/>
        <h:commandButton value="Increment" action="#{counterBean.increment}" /><br/>
        <h:commandButton value="Decrement" action="#{counterBean.decrement}" /><br/>
        <h:commandButton value="Reset" action="#{counterBean.reset}" /><br/>
        </h:form>
    </h:body>

@ManagedBean
@RequestScoped
public class CounterBean {

    @ManagedProperty(value="#{counter}")
    private Counter counter;

    public String increment(){
        counter.increment();
        return "";
    }

    public String decrement(){
        counter.decrement();
        return "";
    }

    public String reset(){
        counter.reset();
        return "";
    }
}
```

```
@ManagedBean
@SessionScoped
public class Counter {
   private int value=0;

   public void increment() {
      if (value <= 5) {
         value = value + 1;
      } else {
         if (value > 5 && value <= 10) {
            value = value + 2;
         } else {
            if (value > 10) {
               value = value + 3;
            }
         }
      }
   }
   public void decrement(){
      if (value <= 5) {
         value = value - 1;
      } else {
         if (value > 5 && value <= 10) {
            value = value - 2;
         } else {
            if (value > 10) {
               value = value - 3;
            }
         }
      }
   }
   public void reset(){
      value=0;
   }

   public int getValue() {
      return value;
   }

}
```

Question 6. [5 points] {10 minutes}

Describe how we can relate the concept of **Model-View-Controller** to the principles of SCI. Your answer should be about half a page, but should not exceed one page (handwritten). The number of points you get for this question depend on how well you explain the relationship between **Model-View-Controller** and the principles of SCI.