Student ID_____Student Name_____

CS545 Web Applications Architecture and Frameworks
DE Final Exam solution
June 28, 2014

Question 1 [5 points] {10 minutes}

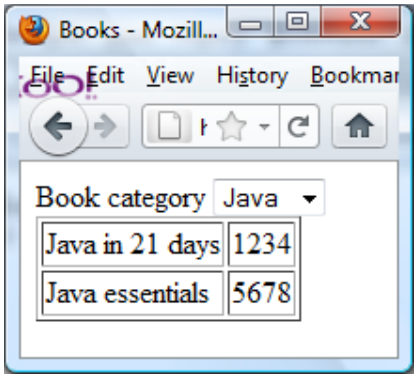
We learned that all HTTP requests for a JSF application are handled by the FacesServlet.

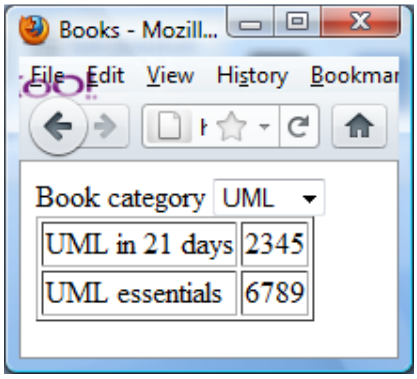
For example, if you sent an HTTP request from the browser with URL: **http://localhost:8080/MyJSFApp/main.xhtml** , then this HTTP request will execute the service() method of the FacesServlet. Explain clearly how the web container knows for which HTTTP requests it should call the service() method of the FacesServlet.

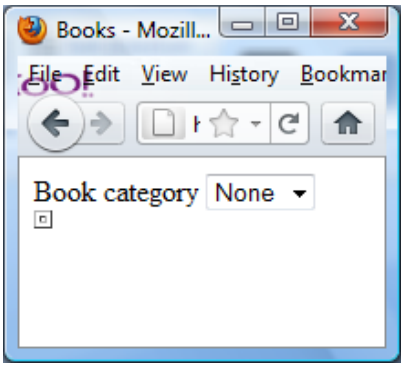
In web.xml we specify which url's should be handled by the FacesServlet

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

Question 2 [20 points] {20 minutes}







Write an JSF application with the following behavior:
The application shows a selectOneMenu control with the values Java, UML and None.
If you select Java, then you see a table showing 2 Java books.
If you select UML, then you see a table showing 2 UML books.
If you select None, then the table is empty.
The first column in the table is the name of the book, and the second column is the ISBN number (in this example just a string containing 4 characters)

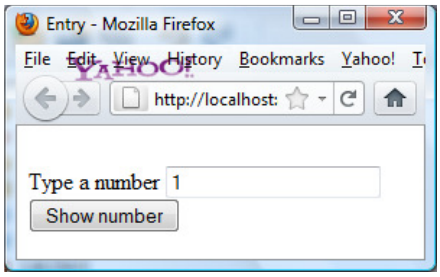
Complete the partial given code such that the application works with the given behavior.
You only have to complete the code for the xhtml file and the managed bean.
For this question you are NOT allowed to use AJAX.
IMPORTANT: do not write getter and setter methods!

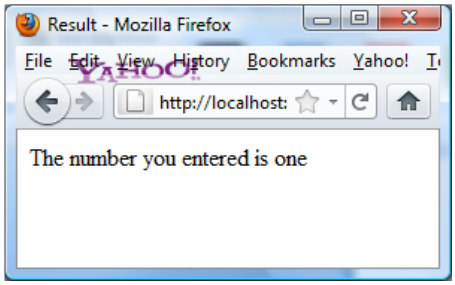
books.xhtml

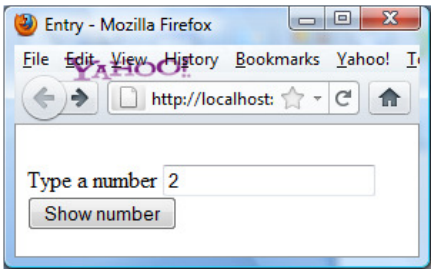
```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core">
    <h:head>
       <title>Books</title>
    </h:head>
    <h:body>
       <h:form>
    Book category
    <h:selectOneMenu value="#{bookbean.selectedBook}"
                valueChangeListener="#{bookbean.changeBook}"
                onchange="submit()">
       <f:selectItems value="#{bookbean.books}"/>
    </h:selectOneMenu>
    <br />
    <h:dataTable value="#{bookbean.booklist}" var="book" border="1" >
       <h:column>
         <h:outputText value="#{book.name}"/>
       </h:column>
       <h:column>
         <h:outputText value="#{book.isbn}"/>
       </h:column>
       </h:dataTable>
       </h:form>
    </h:body>
```

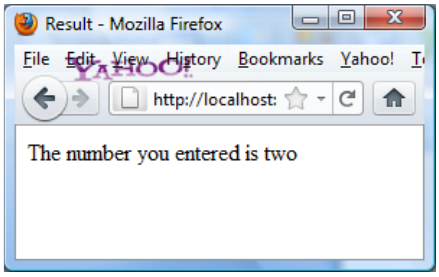
```
import java.util.*;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.event.ValueChangeEvent;
import javax.faces.model.SelectItem;

@ManagedBean
@RequestScoped
public class Bookbean {


   private String selectedBook;
   private Collection<SelectItem> books = new ArrayList();
   private Collection<Book> booklist= new ArrayList<Book>();

   public Bookbean(){
      books.add(new SelectItem("None","None"));
      books.add(new SelectItem("Java","Java"));
      books.add(new SelectItem("UML","UML"));
   }

   public void changeBook(ValueChangeEvent valueChangeEvent) {
      if (valueChangeEvent.getNewValue().toString().equals("Java")){
         booklist.add(new Book("Java in 21 days", "1234"));
         booklist.add(new Book("Java essentials", "5678"));
      } else if (valueChangeEvent.getNewValue().toString().equals("UML")){
         booklist.add(new Book("UML in 21 days", "2345"));
         booklist.add(new Book("UML essentials", "6789"));
      } else {
         booklist.clear();
      }
   }



}
```

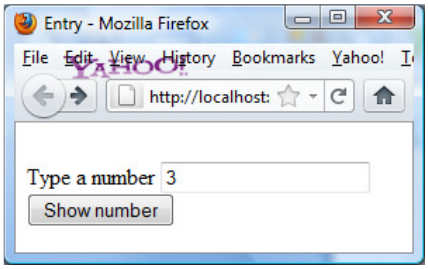
Question 3 Conversion[20 points] {20 minutes}

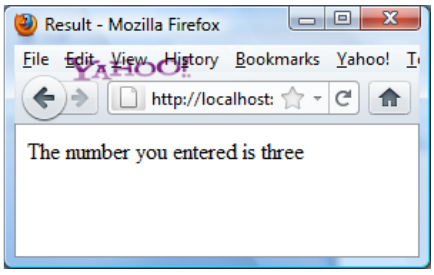


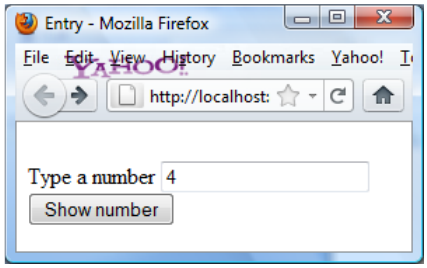


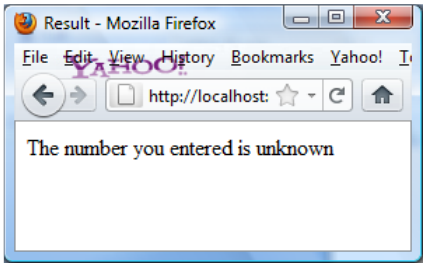












Write a JSF application with the following behavior:
The application consists of 2 pages, an **entry** page and a **shownumber** page.
In the entry page you type in a number.
This number needs to be converted to a String using the following conversion logic:

- Number 1 is converted to the String **one**
- Number 2 is converted to the String **two**
- Number 3 is converted to the String **three**
- All other numbers are converted to the String **unknown**

For this application you have to write your own Converter class.

Complete the partial given code such that the application works with the given behavior. You do not need to worry about validation, you can assume the user will enter a valid number in the entry page. **You only have to complete the code for the xhtml file and the converter class.**

```
@ManagedBean
@RequestScoped
public class Numberbean {
   private String number;
   public String getNumber() {
      return number;
   }
   public void setNumber(String number) {
      this.number = number;
   }
}
```

```
@FacesConverter( value="numberConverter" )
public class NumberConverter implements Converter {

   @Override
   public Object getAsObject(FacesContext context, UIComponent component, String
value) {
      String result = "unknown";
      if (value.equals("1"))
         result = "one";
      if (value.equals("2"))
         result = "two";
       if (value.equals("3"))
         result = "three";
      return result;
    }

   @Override
   public String getAsString(FacesContext context, UIComponent component, Object
value) {
      return value.toString();
   }

}
```

entry.xhtml

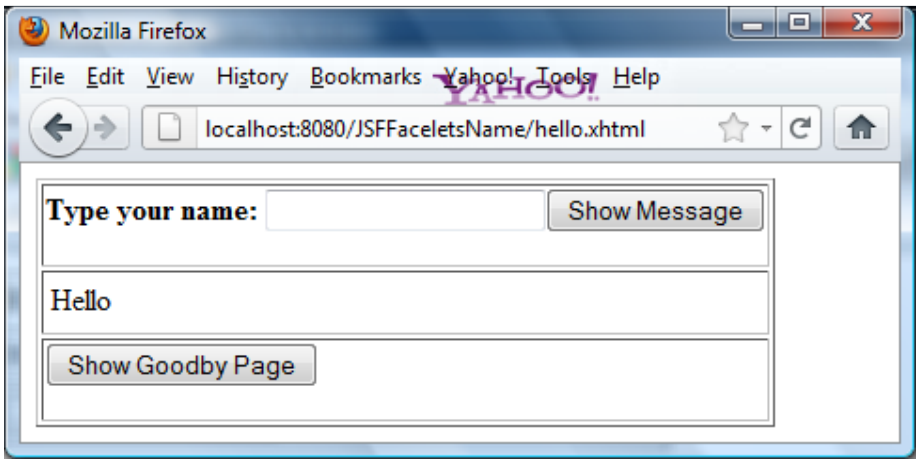
```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core">
    <h:head>
       <title>Entry</title>
    </h:head>
    <h:body>
       <h:form>
          <br/>Type a number

          <h:inputText value="#{numberbean.number}" >
             <f:converter converterId="numberConverter"/>
          </h:inputText>



          <h:commandButton value="Show number"
                           action="shownumber.xhtml"/>
       </h:form>
    </h:body>
</html>
```

shownumber.xhtml

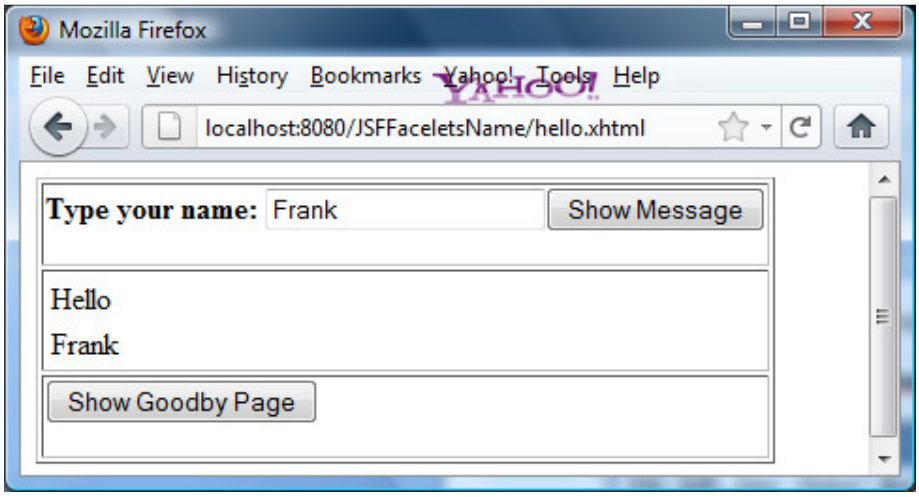
```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
    <h:head>
       <title>Result</title>
    </h:head>
    <h:body>
       The number you entered is

      <h:outputText value="#{numberbean.number}"/>


    </h:body>
</html>
```

Question 4 Facelets [30 points] {35 minutes}
We need to write the following application using facelets:

We have 2 pages: **hello.xhtml** and **goodby.xhtml**

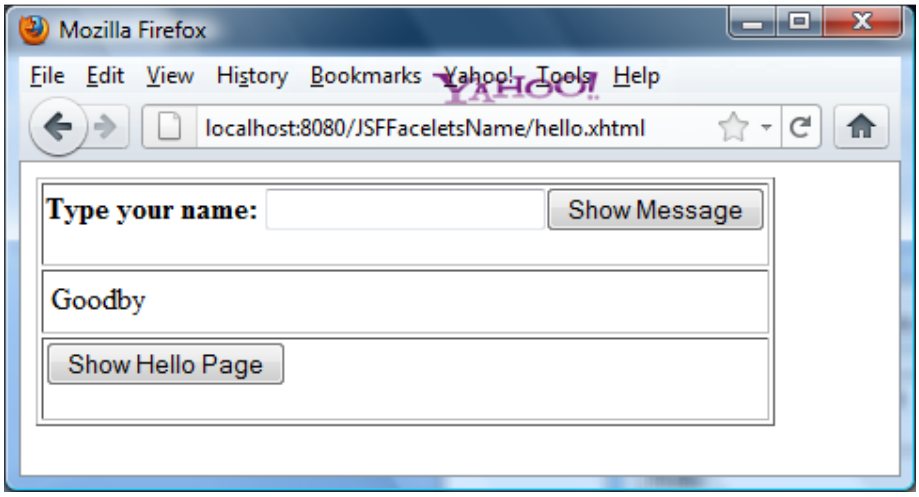


hello.xhtml shows a header at the top, a footer at the bottom and content at the middle the page.

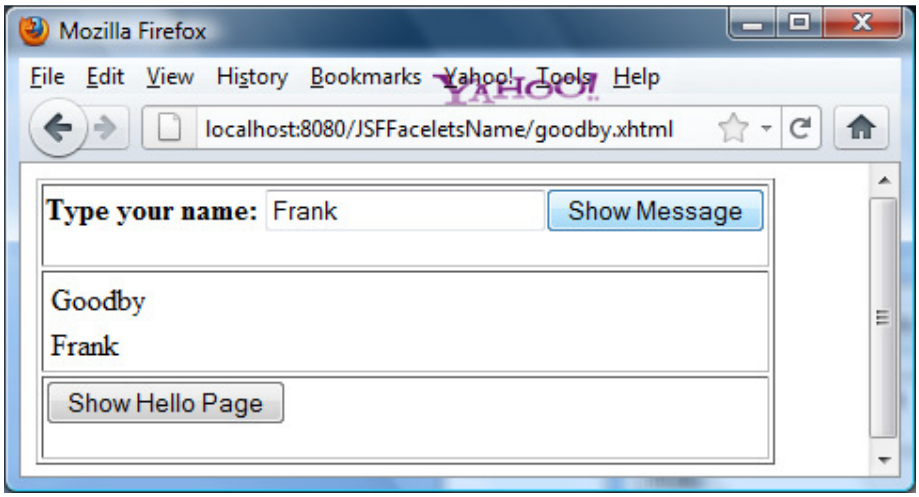


If you type the name "Frank" in the inputtext of the header and click the Show Message button, the content of the page shows the text "Hello Frank"

If you click the "Show Goodby Page" button in the footer, you navigate to **goodby.xhtml**



In **goodby.xhtml**, if you type the name "Frank" in the inputtext of the header and click the Show Message button, the content of the page shows the text "Goodby Frank"



If you click the "Show Hello Page" button in the footer, you navigate to **hello.xhtml**

Implement this application with **facelets**.

Complete the partial given code of the necessary JSF pages such that the application works with the given behavior using facelets. Make sure you add all code and annotations necessary for the correct working of this application.

```
@ManagedBean
@RequestScoped
public class Names {

  private String name;

  public String showMessage(){
     return null;
  }
  public String showHello(){
     return "hello";
  }
  public String showGoodby(){
     return "goodby";
  }
 //getters and setters are not shown
}
```

template.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:ui="http://java.sun.com/jsf/facelets"
     xmlns:h="http://java.sun.com/jsf/html"
     xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">
 <head>
 </head>
 <body>
   <h:panelGrid columns="1" border="1">
     <f:facet name="header">
       <ui:insert name="header" />
     </f:facet>
     <ui:insert name="content" />
     <f:facet name="footer">
       <ui:insert name="footer" />
     </f:facet>
   </h:panelGrid>
 </body>
</html>
```

hello.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">

 <ui:composition template="template.xhtml">
  <ui:define name="header">
   <ui:include src="header.xhtml" />
  </ui:define>
  <ui:define name="content">
   <ui:include src="hellocontent.xhtml" />
  </ui:define>
  <ui:define name="footer">
   <ui:include src="hellofooter.xhtml" />
  </ui:define>
 </ui:composition>
</html>
```

goodby.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">

 <ui:composition template="template.xhtml">
  <ui:define name="header">
   <ui:include src="header.xhtml" />
  </ui:define>
  <ui:define name="content">
   <ui:include src="goodbycontent.xhtml" />
  </ui:define>
  <ui:define name="footer">
   <ui:include src="goodbyfooter.xhtml" />
  </ui:define>
 </ui:composition>
</html>
```

header.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">
 <ui:composition>
    <h:form>
    Type your name:
   <h:inputText id="name" value="#{names.name}"/>
   <h:commandButton value="Show Message"
                action="#{names.showMessage}"/>
   </h:form>
 </ui:composition>
</html>
```

hellofooter.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">
 <ui:composition>
    <h:form>
   <h:commandButton value="Show Goodby Page"
                action="#{names.showGoodby}"/>
   </h:form>
 </ui:composition>
</html>
```

goodbyfooter.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">
 <ui:composition>
    <h:form>
   <h:commandButton value="Show Hello Page"
                action="#{names.showHello}"/>
   </h:form>
 </ui:composition>
</html>
```

hellocontent.xhtml

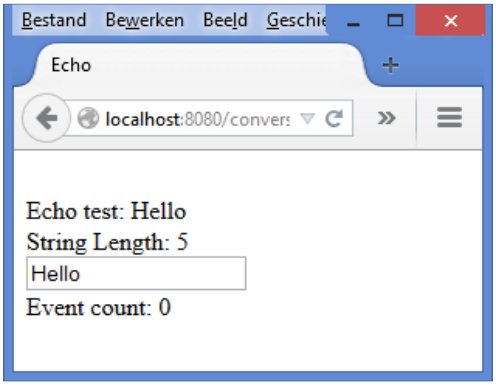
```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">
 <ui:composition>
    <h:panelGrid columns="1" >
   Hello
   <h:outputText id="name" value="#{names.name}"/>
    </h:panelGrid>
 </ui:composition>
</html>
```

goodbycontent.xhtml

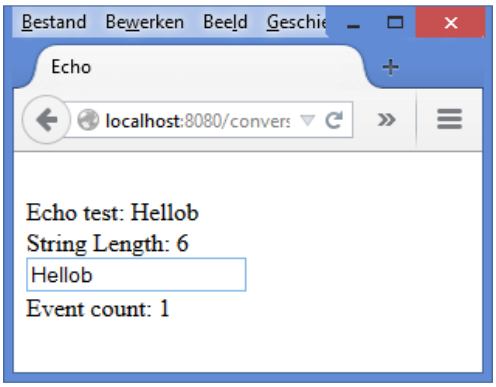
```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core" xml:lang="en" lang="en">
 <ui:composition>
    <h:panelGrid>
   Goodby
   <h:outputText id="name" value="#{names.name}"/>
   </h:panelGrid>
 </ui:composition>
</html>
```

Question 5 AJAX [20 points] {25 minutes}

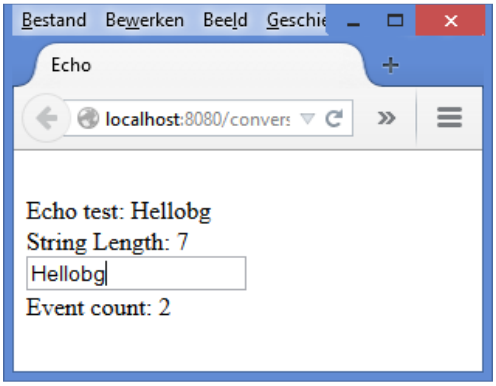
Implement the following application with JSF using AJAX:



When you start the application, you see that the String “Hello” is filled in in the inputText , the same String “Hello” is echoed , the length of the String “Hello” is 5 and event count is 0.



When you type a character in the inputText you see the echo of the string you typed so far, the length of the string you typed so far, and the event count is incremented with 1.



When you type another character in the inputText you see the echo of the string you typed so far, the length of the string you typed so far, and the event count is incremented with 1.

Complete the partial given code of the necessary JSF pages such that the application works with the required behavior. Make sure you add all code and annotations necessary for the correct working of this application.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core">
  <h:head>
    <title>Echo</title>
  </h:head>
  <h:body>
    <h:form >
        <br/>
        Echo test: <h:outputText id="out" value="#{listenBean.hello}"/>
        <br/>
        String Length: <h:outputText id="count" value="#{listenBean.length}"/>
        <br/>
        <h:inputText id="in" value="#{listenBean.hello}" autocomplete="off">
          <f:ajax event="keyup" render="out count eventcount"
              listener="#{listenBean.update}" />
        </h:inputText>
        <br/>
        Event count: <h:outputText id="eventcount" value="#{listenBean.eventCount}"/>
      </h:form>

  </h:body>
</html>
```

```
@ManagedBean
@ViewScoped
public class ListenBean implements Serializable{

  private String hello = "Hello";
  private int length = hello.length();
  private int eventCount = 0;


  public void update(AjaxBehaviorEvent event) {
     length = hello.length();
     eventCount++;
   }

// getter and setter methods are not shown

}
```

Question 6 SCI [5 points] {10 minutes}

Describe how we can relate **JSF custom components** 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 questions depend on how well you explain the relationship between **JSF custom components** and the principles of SCI.