Student ID___________________Student Name______________________________________

Web Applications Architecture and Frameworks DE

Midterm Exam October 25, 2014

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.**

2 blank pages are provided for writing the solutions and/or scratch paper. All 2 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: [10 points] {10 minutes}

A button in JSF is expressed with the following tag:

<h:commandButton value="Press me" action="#{backingbean.doSomething} " />

The <h:commandButton> tag is this example has 2 tag attributes:

1. value
2. action

Give the four possible tag attributes of the **<f:ajax>** tag in JSF and explain what they mean.

Your answer:

- event
    - indicates which event to use to trigger the ajax request
- listener
    - a method that is called on the server side every time the ajax function happens on the client side
- render
    - components to be rendered
- execute
    - Components to be processed on the server

Question 2: [15 points] {15 minutes}

Suppose I have the following files:

template.xhtml:

```
<body>
   <h:panelGrid columns="2" border="1">
     <f:facet name="header">
       <ui:insert name="header" > Default Header </ui:insert>
     </f:facet>
     <ui:insert name="menu" > Default Menu </ui:insert>
     <ui:insert name="content" > Default Content </ui:insert>
   </h:panelGrid>
 </body>
```

header.xhtml:

```
 <ui:composition>
   <h:outputText value="This is header 1" />
 </ui:composition>
```

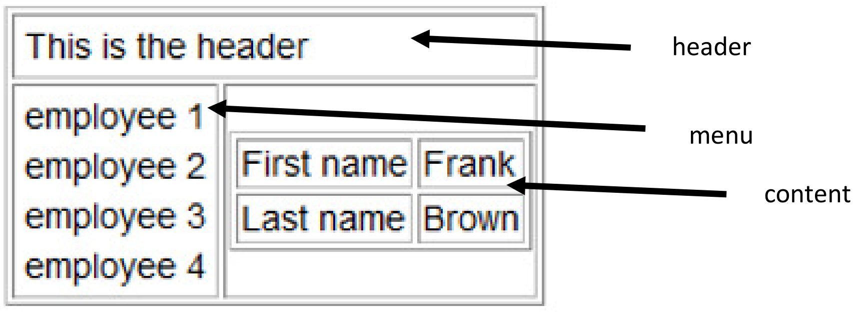
menu.xhtml:

```
<ui:composition>
   <h:panelGrid columns="1">
   <h:outputText value="employee 1" />
   <h:outputText value="employee 2" />
   <h:outputText value="employee 3" />
   <h:outputText value="employee 4" />
   </h:panelGrid>
 </ui:composition>
```

content.xhtml:

```
 <ui:composition>
   <h:panelGrid columns="2" border="1">
     <h:outputText value="First name" />
     <h:outputText value="Frank" />
     <h:outputText value="Last  name" />
     <h:outputText value="Brown" />
   </h:panelGrid>
 </ui:composition>
```

Give the code that I need to write so that I get the following page:



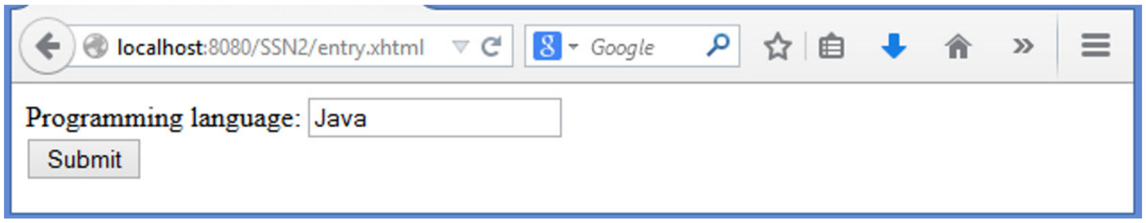
For this exercise you have to use **facelets**, and your have to use the given pages **template.xhtml, header.xhtml, menu.xhtml** and **content.xhtml**.

Your answer:

```
<ui:composition template="template.xhtml">
  <ui:define name="header">
    <ui:include src="header.xhtml" />
  </ui:define>
  <ui:define name="menu">
    <ui:include src="menu.xhtml" />
  </ui:define>
  <ui:define name="content">
    <ui:include src="content.xhtml" />
  </ui:define>
</ui:composition>
```

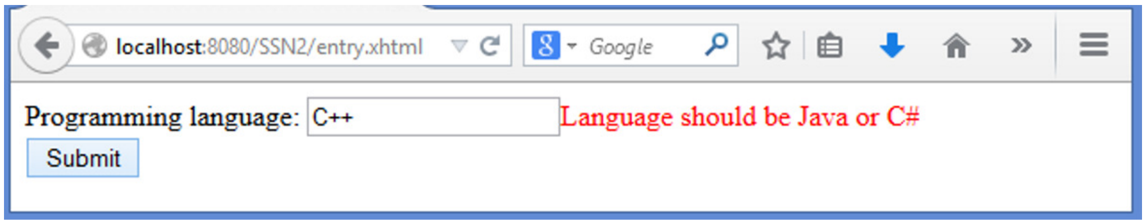
Question 3 Validation [20 points] {20 minutes}

Write a JSF application with the following behavior:



Programming language: Java
Submit

The application asks you to enter a programming language
The only 2 allowed languages are "Java" and "C#"
If you type something different, then you get the following validation message:



Programming language: C++ Language should be Java or C#
Submit

For this application you have to write a Validator component that you can reuse for every inputText where you have to enter a programming language.

Complete the partial given code such that the application works with the given behavior. You have to implement all java code, required annotations and JSF tags that are missing.
IMPORTANT: do not write getter and setter methods!

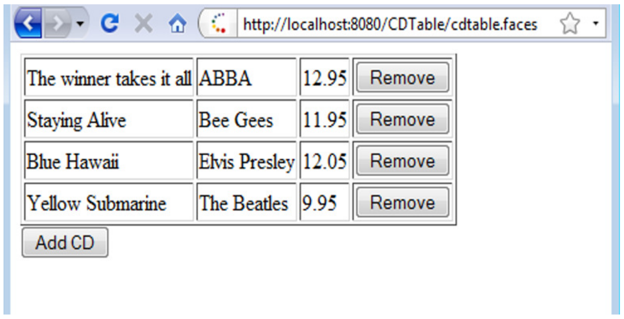
entry.xhtml

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

<body>
      <h:form>
     Programming language:
        <h:inputText id="language" value="#{personBean.language}" >
                  <f:validator validatorId="languageValidator"/>
            </h:inputText>
            <h:message style="color: red" for="language"/>

            <br />
            <h:commandButton action="#{personBean.submit}" value="Submit" />
      </h:form>
</body>
</html>
```

```
@ManagedBean
@RequestScoped
public class PersonBean {
      private String language;

      public String submit(){
            return "";
      }
   // gettter and setters are not shown
}
```

```
      @FacesValidator("languageValidator")
      public class LanguageValidator implements  Validator{
          public void validate(FacesContext facesContext, UIComponent uIComponent,
Object object) {

              String language = (String)object;

              if ((!language.equals("Java") || language.equals("C#"))){
               FacesMessage message = new FacesMessage();
               message.setSummary("Language should be Java or C#");
               message.setSeverity(FacesMessage.SEVERITY_ERROR);
               throw new ValidatorException(message);
              }
          }
      }
```

Question 3 [25 points] {35 minutes}
Write a JSF application with the following behavior:

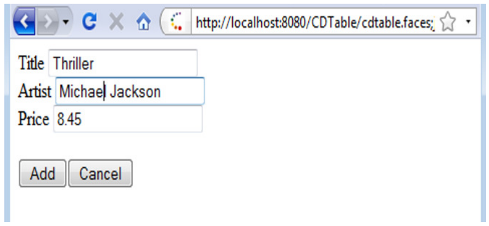


http://localhost:8080/CDTable/cdtable.faces

The winner takes it all	ABBA	12.95	Remove
Staying Alive	Bee Gees	11.95	Remove
Blue Hawaii	Elvis Presley	12.05	Remove
Yellow Submarine	The Beatles	9.95	Remove

Add CD

The application shows a list of CD's in a table. If you click the Remove button, that particular CD will be removed from the list.
If you click the Add CD button, then this page will be shown:



http://localhost:8080/CDTable/cdtable.faces;

Title	Thriller
Artist	Michael Jackson
Price	8.45

When you enter the CD information, and you click the Add button, then you return to the previous CD table page, and you see that this new CD is added to the list of CD's.

Complete the partial given code such that the application works with the given behavior. You have to implement all java code, required annotations and JSF tags that are missing.
IMPORTANT: do not write getter and setter methods!

cdcollection.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>CD collection</title>
    </h:head>
    <h:body>
    <h:form>
     <h:dataTable value="#{ cdtable.cdlist.cdlist}" var="cd" border="1" >
        <h:column>
          <h:outputText value="#{cd.title}"/>
        </h:column>
        <h:column>
          <h:outputText value="#{cd.artist}"/>
        </h:column>
        <h:column>
          <h:outputText value="#{cd.price}"/>
        </h:column>
        <h:column>
           <h:commandButton value="Remove" action="#{ cdtable.removeCD(cd)}"/>
        </h:column>
        </h:dataTable>
        <h:commandButton value="Add CD"
                 action="addcd"/>
    </h:form>
    </h:body>
</html>
```

__addcd.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>Add new CD</title>
   </h:head>
   <h:body>
     <h:form>
     Title
     <h:inputText value="#{addcd.title}"/>
     <br />
     Artist
     <h:inputText value="#{addcd.artist}"/>
     <br />
     Price
     <h:inputText value="#{addcd.price}"/>
     <br />
     <br />
     <h:commandButton value="Add"
                action="#{addcd.add}"/>
     <h:commandButton value="Cancel" type="submit"
                action="cdcollection"/>
   </h:form>
   </h:body>
</html>
```

__CD.java__

```
public class CD {
  private String title;
  private String artist;
  private double price;

   //getters and setters are not shown
}
```

Cdtable.java

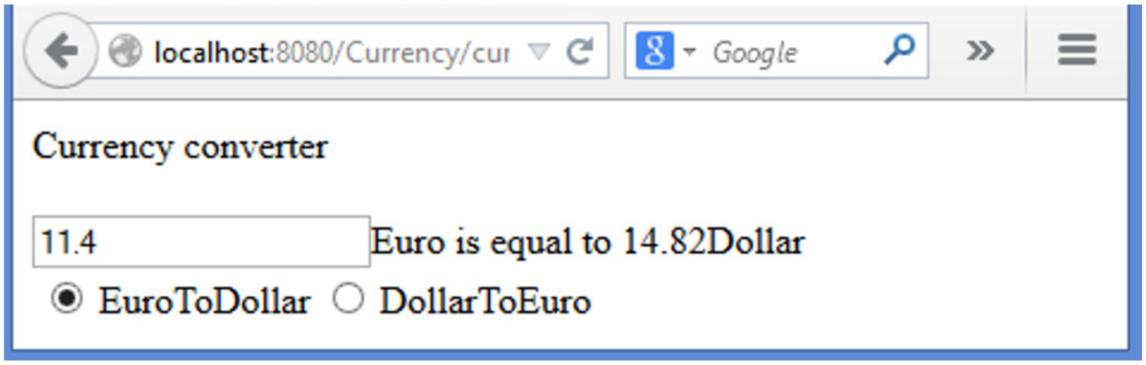
```
@ManagedBean
@RequestScoped
public class Cdtable {


   @ManagedProperty(value="#{ cDCollection }")
    CDCollection cdlist;


    public String removeCD(CD cd) {
      cdlist.removeCD(cd.getTitle());
      return "";
    }

    //getters and setters are not shown (do not write getter and setter methods)
}
```

Addcd.java

```
@ManagedBean
@RequestScoped
public class Addcd {


    private String title = "";
    private String artist = "";
    private double price ;

    @ManagedProperty(value="#{ cDCollection }")
    CDCollection cdlist;

    public String add() {
       CD newCD = new CD(title, artist,price);
       cdlist.addCD(newCD);
       return "cdcollection";
    }


      //getters and setters are not shown (do not write getter and setter methods)
}
```

```
@ManagedBean
@SessionScoped
public class CDCollection implements Serializable{
   private Collection<CD> cdlist = new ArrayList();

   public CDCollection() {
      cdlist.add(new CD("The winner takes it all ", "ABBA", 12.95));
      cdlist.add(new CD("Staying Alive ", "Bee Gees", 11.95));
      cdlist.add(new CD("Blue Hawaii ", "Elvis Presley", 12.05));
      cdlist.add(new CD("Yellow Submarine ", "The Beatles", 9.95));

   }

   public void addCD(CD cd) {
      cdlist.add(cd);
   }

   public void removeCD(String title) {
      Iterator iter = cdlist.iterator();
      while (iter.hasNext()) {
         CD cd = (CD) iter.next();
         if (cd.getTitle().equals(title)) {
            iter.remove();
         }
      }
   }



}
```

Question 4 [25 points] {30 minutes}
Write a JSF currency converter application with the following behavior:

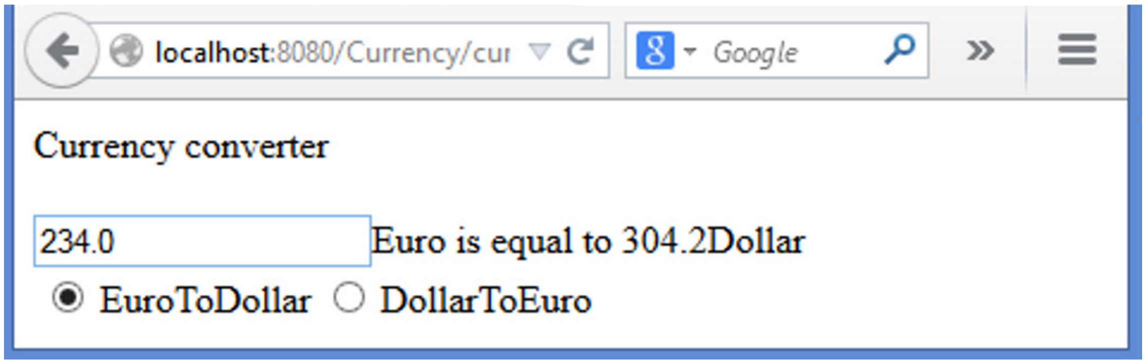


Currency converter

11.4 Euro is equal to 14.82Dollar

● EuroToDollar ○ DollarToEuro

When you enter the value 11.4 in the inputText field and hit return, then the converter shows that 11.4 euro is equal to 14.82 dollar.

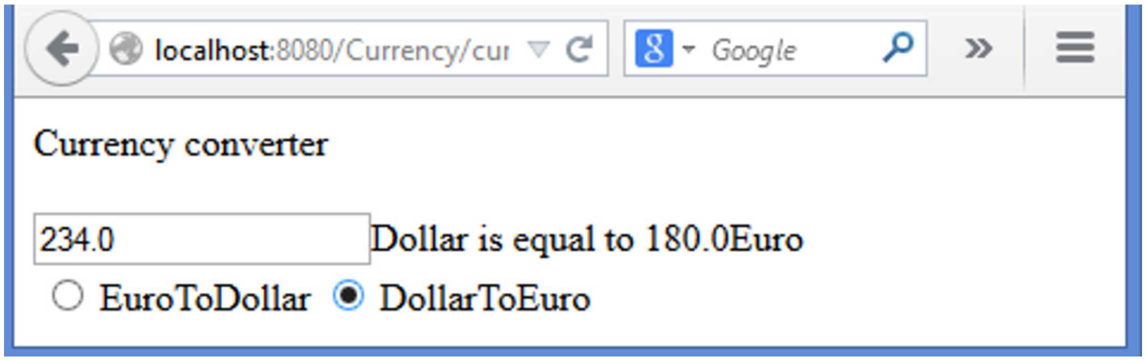


Currency converter

234.0 Euro is equal to 304.2Dollar

● EuroToDollar ○ DollarToEuro

When you enter the value 234.0 in the inputText field and hit return, then the converter shows that 234.0 euro is equal to 304.2 dollar.



Currency converter

234.0 Dollar is equal to 180.0Euro

○ EuroToDollar ● DollarToEuro

When you select the DollarToEuro radio button, then the converter shows that 234.0 dollar is equal to 180.0 euro.
Complete the partial given code such that the application works with the given behavior. You have to implement all java code, required annotations and JSF tags that are missing.

For this exercise you may NOT use AJAX functionality.
The conversion rate is: 1 euro = 1.3 dollars.
IMPORTANT: do not write getter and setter methods!

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

<body>
     <h:form>
     Currency converter <br/><br/>
        <h:inputText  value="#{converter.input}"
valueChangeListener="#{converter.changeValue}"
                            onchange="submit()"/>
        <h:outputText value="#{converter.inputSymbol}"/>
        <h:outputText value=" is equal to "/>
           <h:outputText value="#{converter.output}"/>
        <h:outputText value="#{converter.outputSymbol}"/>
           <br />
             <h:selectOneRadio value="#{converter.selection}"

valueChangeListener="#{converter.changeSelection}"
                            onchange="submit()">
            <f:selectItem itemValue="EuroToDollar"
itemLabel="EuroToDollar"/>
            <f:selectItem itemValue="DollarToEuro"
itemLabel="DollarToEuro"/>
          </h:selectOneRadio>
     </h:form>
</body>
</html>
```

```java
@Named("converter")
@SessionScoped
public class ConverterBean implements Serializable{
      private double input;
      private double output;
      private String inputSymbol;
      private String outputSymbol;
      private String selection = "EuroToDollar";


    public ConverterBean() {
      inputSymbol="Euro";
      outputSymbol="Dollar";
      }

      public void changeValue(ValueChangeEvent valueChangeEvent) {
         double newvalue = (Double) valueChangeEvent.getNewValue();
         input=newvalue;
         convert();
     }

    public void changeSelection(ValueChangeEvent valueChangeEvent) {
         String newvalue = (String) valueChangeEvent.getNewValue();
         if (newvalue.equals("EuroToDollar")){
            inputSymbol="Euro";
            outputSymbol="Dollar";
       }
       else{
            inputSymbol="Dollar";
            outputSymbol="Euro";
       }
         convert();

     }

    public void convert(){
      if (inputSymbol.equals("Euro")){
            output = input * 1.3;
      }
      else{
            output = input / 1.3;
      }
    }

}
```

Question 5. SCI [5 points] {10 minutes}

Describe how we can relate the concept of **JSF facelets** 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 **JSF facelets** and the principles of SCI.

Your answer: