Student ID_____Student Name_____

Web Applications Architecture and Frameworks DE

Final Exam June 25, 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.**

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.

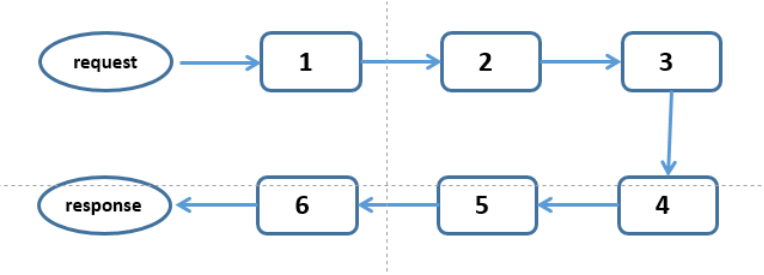
Write your **name and student id** at the top of this page.

All your answers have to be written in the designated areas on the exam paper. If you make a mistake, you can use the extra blank pages.

Write clearly. If I cannot read it, I cannot give any points.

Question 1: [10 points] {10 minutes}

The JSF lifecycle consists of 6 phases:



a. Give the name of every phase and in one sentence describe what JSF does in this phase.

Phase name	Short description of what JSF does in this phase
1.	
2.	
3.	
4.	
5.	
6.	

b. JSF splits the JSF lifecycle into 2 parts: execute and render

Give the number(s) of the phases that belong to the execute part

Give the number(s) of the phases that belong to the render part

Question 2: [10 points] {10 minutes}

Given are 2 code snippets:

Snippet 1:

```
<h:form
<h:selectOneMenu value="#{bean.value}" valueChangeListener="#{bean.changeListener}">
   <f:selectItems ... />
</h:selectOneMenu>
<h:/form>
```

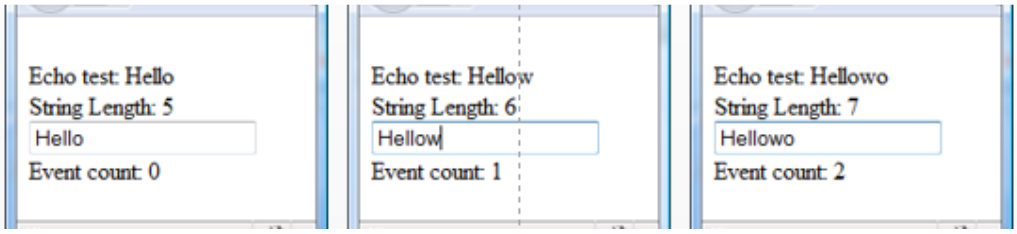
Snippet 2:

```
<h:selectOneMenu value="#{bean.value}">
   <f:selectItems ... />
   <f:ajax listener="#{bean.ajaxListener}" />
</h:selectOneMenu>
```

Explain clearly the difference in behavior that you will observe when you use snippet 1 or snippet 2.

Question 3: [10 points] {15 minutes}

Suppose we want the following behavior of a JSF application:



Whenever you type a character in the input field, the page shows the following info:

- The first line just shows the same string as is shown in the input field
- The second line shows the length of the string as is shown in the input field
- The bottom line shows how many times a new character is typed in the input field

For this exercise you have to use AJAX. Modify the given code (on the next page) so that we get the given behavior

```
<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}" >




        <br/>
        Event count: <h:outputText id="eventcount" value="#{listenBean.eventCount}"/>
      </h:form>



@ManagedBean
@ViewScoped
public class ListenBean implements Serializable{
  private String hello = "Hello";
  private int length = hello.length();
  private int eventCount = 0;




 …
}
```

Question 4: [10 points] {15 minutes}

Suppose we have a JSF form with an inputText like this
<h:inputText *value="#{studentForm.student}"* >

The form expects that students are entered in the format ***"name-age"***. Examples are ***John-66*** or ***Frank-22***

The student in the managedBean is of type Student:

```
@ManagedBean
@RequestScoped
public class StudentForm {
   private Student student;
... //constructor, getters and setters are not shown
}

public class Student {
  private String name;
  private Integer age;
... //constructor, getters and setters are not shown
}
```

Write a JSF custom converter that converts the string that is typed in the inputText to a Student class, and also the other way around (from Student class to String)

Question 5: [10 points] {10 minutes}

JSF supports different ways to apply validation to your application. Give 3 different validation techniques that are supported by JSF

1.
2.
3.

Question 6: [10 points] {15 minutes}

a. Give 3 different ways of how we can implement navigation in JSF. Give an example of each.

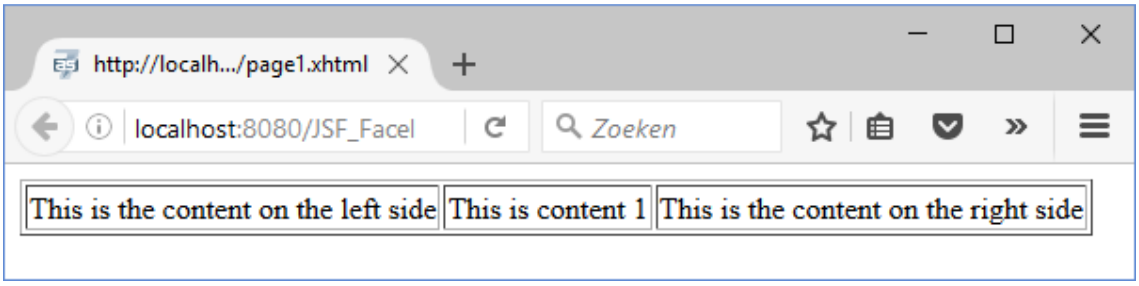
1.
2.
3.

b. What is the default navigation implementation of JSF, **forward** navigation or **redirect**?

c. Give an example of how we can change the default navigation.

Question 7: [10 points] {15 minutes}

Write the following application with JSF and **facelets**:
page1.xhtml



page2.xhtml

Every page we have to make is divided into 3 parts: a left part, a middle part and a right part.
We need to use **facelets** to make it easy to give every page this same structure.
Given are the following files:

left.xhtml:
```
<ui:composition>
  <h:outputText value="This is the content on the left side"/>
</ui:composition>
```

right.xhtml:
```
<ui:composition>
  <h:outputText value="This is the content on the right side"/>
</ui:composition>
```

content1.xhtml:
```
<ui:composition>
      This is content 1
</ui:composition>
```

content2.xhtml:
```
<ui:composition>
      This is content 2
</ui:composition>
```

Write **all** the necessary files that are missing.
Your implementation should follow the following requirements:

1. You are **not** allowed to use HTML, JSP's or servlets.
2. You need to use **facelets**
3. **For web pages, you only need to write the code between the <body> and </body> tags. Do not write the code for namespaces**

page1.xhtml:

page2.xhtml:

Question 8: [10 points] {10 minutes}

Suppose we have the following CD class:

```
public class CD {
  private String title = "";
  private String artist = "";
  private double price = 0;
  …
}
```

And the following list of CD's in a managed bean:

```
public class TheManagedbean
    private Collection<CD> cdList=new ArrayList<CD>();
```

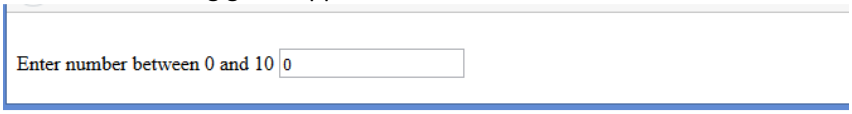
And we want to show the content of the cdList in a JSF table like this:

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

Write the part of code we need in the xhtml page that shows the content of the cdList like the example given above:

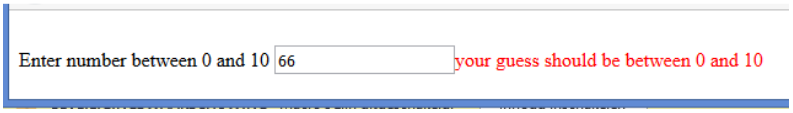
Question 9: [15 points] {20 minutes}

Write the following guess application with JSF:



The applications ask to enter a number between 0 and 10. The number that needs to be guessed is hard coded at the number 5.
When you type a number in the input field, the application will show if this number is too high, too low or correct:

When you enter an incorrect input, the application will show this:



Add the missing code on the next page.
Your implementation should follow the following requirements:

1. You are **not** allowed to use HTML, JSP's or servlets.
2. Do **not** write getter and setter methods
3. **For web pages, you only need to write the code between the <body> and </body> tags. Do not write the code for namespaces**

<h:form >

</h:form>

public class GuessBean {

}