Three hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Building Web Applications

Date: Friday 29th May 2009
Time: 09:45 – 12:45

Answer Question 1 from Section A, and TWO questions out of the three from Section B.
Each question carries 20 marks.
This is a CLOSED book examination

The use of electronic calculators is NOT permitted
Section A

1. Answer each of the following questions concisely. **Each carries two marks.**

   a) What are the three main stages in which a servlet handles a HTTP request? What are the fundamental differences between these three stages, and how should they be reflected in the design of the servlet?

   b) What is the role of the `init` function? How and when is it called exactly? What is the role of the `destroy` function? How and when is it called exactly?

   c) Which HTTP command causes the call to the `doGet` function in a Java servlet? What are the two parameters of this function and what are their respective types and roles?

   d) Describe the form of parameters in a HTTP command, and how parameter values are obtained within the servlet processing them.

   e) Briefly explain how HCI logic is separated from other concerns in the generic Web application design illustrated by the program Dictionary1 of Exercise Ex04.

   f) The program RsrDB of Exercise Ex05 contains two errors. One occurs within the execution of the `init` function. Explain the cause of this error and how it is corrected in the program DictionaryDB of Exercise Ex06.

   g) The second error of RsrDB is due to the semantics of the SQL DELETE command, used to implement the operation Rem of the Evolving Dictionary Model (EDM). Which precondition of Rem does not apply to DELETE? How did this discrepancy lead to the second error of RsrDB, what is the error exactly, when does it manifest itself when using the program and how is it rectified in DictionaryDB?

   h) Briefly describe the three main stages involved in using JDBC, and their respective objects.

   i) Explain the notion of a `hidden field` in HTML. In the program DictionaryST, hidden fields are used for three purposes (two jointly with Dictionary1 and DictionaryDB). Describe these three purposes.

   j) Briefly describe the Java exception handling mechanism. Why is it particularly important in applications using JDBC?

   (20 marks in total)
Section B – Answer TWO questions out of three

2. a) Describe in outline the concept of a JSP page, by comparison with an ordinary HTML page, and indicate what the acronym 'JSP' stands for. (4 marks)

b) Consider the following two simplified excerpts from the program DictionaryJSP. Explain the purpose of each excerpt and the features highlighted in boldface. In addition, reconstruct the original segment of code delimited here by `<%//Gap` and `//Gap%>`, according to the comment and using the three method calls `errs.MaxNumErrs(i)`, `errs.IsMb(i)` and `errs.ErrMes(i)`. (6 marks)

Excerpt 1 from B1Page2Gen.java:
```java
public static void DisplayResultV01(HttpServletRequest req, HttpServletResponse res, String com, B0ErrorSet errs)
// Return result of non-query operation.
{
    RequestDispatcher rd = req.getRequestDispatcher("../Page2.jsp");
    req.setAttribute("com", com);
    req.setAttribute("errs", errs);
    rd.forward(req, res);
} // public static void DisplayResultV01()
```

Excerpt 2 from Page2.jsp:
```jsp
<% String com = (String)request.getAttribute("com");
    B0ErrorSet errs = (B0ErrorSet)request.getAttribute("errs");
%>
<HTML> <HEAD> <TITLE> . . . </TITLE> </HEAD> <BODY>
<% if (errs.NumErrs() == 0) {
    <H2>Operation <%=com%> successful.</H2>
}%
else {
    <H2>Operation <%=com%> failed. Errors:</H2>
%}
<% // Display each error found as a H2 heading
    // . . .
%>
</BODY></HTML>
```

(Question 2 continues on the following page)
c) i) Briefly describe the four main stages of the formal method of software development illustrated by the construction of the program Dictionary0 (Exercise Ex03). One stage has two sub-stages, to be included in your description. Draw a diagram to represent the partial ordering of these stages and sub-stages, and explain this ordering.

ii) The program Dictionary0 started with the mathematical model ‘EDM’ ('Evolving Dictionary Model'). The definition of possible states is reproduced here:

\[
\begin{align*}
\text{Words} & : \text{Set} \\
\text{Definitions} & : \text{Set} \\
D & : \text{Dictionary} \\
D & : \text{Words} \to \text{Definitions}
\end{align*}
\]

This was supplemented with the following definitions.

\[
\begin{align*}
\text{DDom} & = \text{Words} & \text{Domain of } D \\
\text{DCod} & = \text{Definitions} & \text{Codomain of } D \\
\text{DGr} & \subseteq \text{DDom} \times \text{DCod} & \text{Graph of } D \\
\text{DDef} & \subseteq \text{DDom} & \text{Definition domain of } D \\
\text{DRan} & \subseteq \text{DCod} & \text{Range of } D
\end{align*}
\]

Explain this model, excluding the definitions.

iii) Using any appropriate definition provided, informally describe the functionality condition of the model and indicate to which stage of your answer to (i) it belongs.

iv) The model was implemented in Java using the variables:

```
int n;  String word[];  String definition[]
```

Formally express the functionality condition of your answer to (iii) as it applies to this representation, in a form as close to Java as possible.

(5 marks)

d) i) The EDM model involved a number of operations including a function to insert a new entry into the dictionary. Its Java representation is `void Ins(String w, String d)`. One component of the formal specification of this function is `due to` the functionality condition mentioned in part (c). Give this component and indicate to which stage of your answer to part (c)(i) it belongs.

ii) The formal specification of the component referred to in (d)(i) affects the actual Java coding in two ways. Describe each of them and how they relate to each other in the light of your answer to part (c)(i).  

(5 marks)
3. a) i) Describe the paradigm Model View Controller (MVC).
   ii) Explain why MVC is an important concept of software design in general and in particular for Web applications.
   iii) Describe the server-side basic processing cycle associated with the HTTP protocol, and show how the design principle underpinning MVC follows from this cycle. (6 marks)

   b) i) Briefly explain the feature of the HTTP protocol causing the problem of session tracking and the reason for this feature.
   ii) Explain the problem of session tracking in detail. In your analysis allow for the various circumstances that may affect different Web applications with respect to functionality and the expected number of simultaneous users.
   iii) Describe the general mechanism at the basis of the various methods of session tracking, and at least one method of implementing this mechanism. (6 marks)

   c) The program DictionaryST of Exercise Ex09 required extending the program Dictionary1 with a 'naive' implementation of the session tracking mechanism referred to in part (b)(iii). The following is the result of extracting various segments of Java from several classes of DictionaryST and jumbling them up. This is followed by specifications of these excerpts in the form of short headings. For each code segment, copy its heading and restore the full original text. N.B. Identical lines are shown once only in the table.

(Question 3 continues on the following page)
(Question 3 continues from the previous page)

```java
+ curST.htmlSesId()
+ curST.getCurrentSesId()
boolean curSessionDefined;
System.err.println("My ERROR: Unable to get current session");
String htmlSesId()
static B0NaiveSessionTracking curST = null;
D = curST.getCurrentSession().getDic1();
+ currentSesId + ""> 
+ "<H3>DictionaryST. Session "
+ " </H3> 
"curSessionDefined = curST.getSessionFromClientOrNew(req);
if (currentSesId == null)
if (curSessionDefined)
curST = new B0NaiveSessionTracking();
} // String htmlSesId()
else
return " <INPUT TYPE="hidden" NAME="sesId" VALUE=""
return;
return "";
[
]

Specifications:

**From A0DictionaryST:**

// Declare current naive session tracking container object curST

**From A0DictionaryST.init:**

// Create curST

**From A0DictionaryST.doGet:**

// Define or redefine current session

// Redefine dictionary D for the duration of the current request
// handling if possible, else log an error message and return

**From B1Page1Gen.DisplayPage, as part of html generation:**

// Display current session index as H3 heading

**From B1Page1Gen.DisplayPage, within the generation of each form:**

// Insert session index using curST

**From B0NaiveSessionTracking: text of function htmlSesId()**

// Return HTML hidden field giving currentSesId if defined,
// "" else. To be used in every form of a generated page.
4. a) Give an illustrated overview of JSF and Visual JSF, addressing the following issues:
   i) Indicate what the acronym 'JSF' stands for.
   ii) Outline the overall aim of JSF from your experience of using it.
   iii) Describe how a Web page is constructed using JSF, the three editor views on it and what object exactly is rendered by each view. Hint: there are two source code objects so rendered.
   iv) Indicate the respective roles of the two source code objects mentioned in (iii). (6 marks)

b) Describe the three beans at the heart of JSF. Outline the role of each bean and how they relate to the code associated with the pages. (6 marks)

c) In the program JSF_Book of Exercise Ex08, pages 2 and 3 respectively give two views on the table ‘Book Copies’ recording book copy loans to members. Each record of this table contains a field ‘BOOK_ID’ and a field ‘BORROWER_ID’. In each page, data entry by the user for these two fields is guaranteed to conform to the relevant entity referential integrity constraint. Explain what this conformance means and how it is achieved. (4 marks)

d) Consider the various servlet and JDBC features explicitly Java-coded in the exercise programs Dictionary1, RsrDB, DictionaryDB, DictionaryJSP and DictionaryST. Many of these features and others are automatically generated by JSF. Give as complete a list of such features as possible, and briefly comment on the advantages and dangers of using frameworks like JSF instead of direct Java coding. (4 marks)

END OF EXAMINATION