Web Applications

Date: Thursday 15th May 2008
Time: 14:00 – 17:00

Please answer Question 1 from SECTION A and TWO questions from the THREE provided in SECTION B

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) Contrast static and dynamic pages in Web applications. Give a full description of the difference between a Web application that returns only static pages, and one capable of returning dynamic pages.
   b) Give the definition of a Java servlet. Describe how a servlet relates to the Web server invoking it and as a result of which events it is (1) loaded and (2) executed?
   c) 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?
   d) Which HTTP command causes the call to the doGet function? What are the two parameters of this function and what are their respective types and roles?
   e) What are the three main stages in which a servlet handles a HTTP request? What are the fundamental differences between these three stages?
   f) What is program design, what decisions does it involve and what key design guiding principles can you mention?
   g) In accordance with your answers to (e) and (f), how should the three stages of (e) be reflected in the design of the servlet?
   h) Describe the main features of a consistent HTML page, covering specifically:
      i) The notion of an element
      ii) The notion and form of a tag, and the types of tags
      iii) How elements relate to one another and the logical structure this imparts to the page
   i) Describe the form and role of an attribute in a consistent HTML page and the types of attributes.
   j) Define the notion of a protocol and give the main features of the HTTP protocol.
   k) Define the notion of a relative URL. Explain how a relative URL is used by a Web browser and how this feature affects servlet programming.
   l) Explain the problem of session tracking and indicate which feature of HTTP causes this problem. Describe the principles common to the various methods of session tracking, including security as a factor.

(Question 1 continues on following page)
(Question 1 continues from previous page)

m) What is the effect of declaring a feature of a Java class as static? How do static features affect multiple instances of a class? How can a static feature be accessed from outside a class C without using an instance of C?

n) What is a program thread? How are threads used in servlets? What are the resultant difficulties and how can they be resolved?

(28 marks)
Section B

2. a) Briefly define the following notions. Indicate how they relate to one another and consequently in which (partial) order they should be established:

i) An assertion in a computer program
ii) A program invariant
iii) A formal function specification in terms of input, pre-conditions, output and post-conditions
iv) The use (or call) of a function and the implementation of a function.

(4 marks)

b) Consider a Java class `ListW` designed to store the words occurring in a document and for each word its multiplicity i.e. the number of its occurrences in the document. The words and their multiplicities must be represented in main store and in ascending order, using the following Java variables:

final int Max = 10; // Maximum number of entries in the list
int n; // Number of entries in the list
String[] word = String[Max]; // Words represented
int[] mult = String[Max]; // Multiplicities of the words represented

Formalize the following program invariants in terms of these declarations and in a form as close to Java as possible:

Inv 1 Each represented word $w$ must be stored in the array `word` at a position $i$ in the range 0 to $n-1$ and its multiplicity at the same position in `mult`.
Inv 2 A word must be a non-empty String.
Inv 3 A word may occur at most once in the list.
Inv 4a The number of represented words cannot be negative.
Inv 4b The number of represented words cannot exceed the capacity of the list.
Inv 5 The words must be stored in ascending order. (4 marks)

c) Give a precise specification of an appropriate function `Ins` such that for any String $w$, a call `Ins(w)` can be used to record any one occurrence of $w$ found in the targeted document. As part of the specification list all the pre-conditions needed to preserve your formalized invariants, indicating which invariant is preserved by each pre-condition given. For any variable $x$ varying in `Ins`, let $x$ denote the value of $x$ at the start of `Ins` and $x'$ the value at the end of `Ins`. (4 marks)

(Question 2 continues on following page)
d) It is sometimes argued that a function should have no pre-conditions, as a factor of program safety and robustness. As a general rule, this is a misconception.

i) Indicate when it is appropriate, or possibly inevitable, to impose pre-conditions on a function.

ii) Show that if we consider any segment of code rather than just functions, the necessity for appropriate pre-conditions does become absolute, and as a result, avoidance of pre-conditions is a fallacious pursuit. (4 marks)
3. a) Describe the design principles underpinning the variant Java implementation Dictionary1 of the Evolving Dictionary Model (EDM), as a general framework for Web applications. In particular:

i) Describe the purposes of the four main categories A, B, C and D of classes involved.

ii) Describe how the Web pages of the application are related to category B1 classes in outline.

iii) Explain how this design supports the *scalability* of the application and any other important criterion of good design for Web applications.

(4 marks)

b) i) Illustrate your answer to part (a) by explaining the role of the following (simplified) program components. In particular explain the difference between a command function and the corresponding basic Dictionary operation.

ii) The excerpt below from class C0Dictionary1 contains several errors against the EDM specification and the data structure specified for the implementation. Identify, explain and correct these errors.

```java
// From class A0Dictionary1:
C0Dictionary1 D = new C0Dictionary1(Max); // ...
public void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException
{
    String Page;                             // Page number
    Page = req.getParameter("Page");
    if (Page.equals("0"))
        B1Page0Proc.ProcReq(req, res);
    else if (Page.equals("1"))
        B1Page1Proc.ProcReq(req, res, D); // ...
}

// From class B1Page1Proc:
public static void ProcReq(HttpServletRequest req,
    HttpServletResponse res, C0Dictionary1 D) // ...
{
    String Com;                        // Command http parameter
    Com = req.getParameter("Com");
    if (Com.equals("Emp"))
        ComEmp(D, Com, req, res);
    else if (Com.equals("Ins"))
        ComIns(D, Com, req, res);
    else if (Com.equals("Rem"))
        ComRem(D, Com, req, res);   // . . .
}

// From class C0Dictionary1:
public void Rem(String w, String d)
    // Rem(w, d) in D
    // Initial conditions
    // w, d are non-empty strings
    // for all i with 0 <= i < n (!word[i].equals(w))
    // n < Max
```

(4 marks)

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

c) Implement the function ComRem, after due correction of the specification of the function Rem. (4 marks)

d) Implement the basic operation Rem, after due correction of its specification, in the ‘minimalist’ manner allowed by the correct pre-conditions. (4 marks)
4. a) Given an appropriate driver, the use of JDBC involves three distinct stages:

i) Establishing a ‘connection’
ii) Creating and executing a ‘statement’
iii) Processing any result of the execution of the ‘statement’

Briefly describe these three stages. (4 marks)

b) The following table contains a number of four simplified and jumbled up Java segments taken from the JDBC application RsrDB. Reconstruct the original text including the original line numbers and outline its behaviour by additional line comments e.g. “// Segment 3: start of function Emp()”. Make sure the four original segments are well delimited by such line comments. Some non-essential lines are missing; do not attempt to recreate them, just re-order the given text. Some given lines are needed several times.

```
1 public class RsrDB extends HttpServlet {
2     } // init()
3     private Connection connection; // etc.
4     rs = statement.executeQuery("SELECT * FROM Res");
5     Statement statement = connection.createStatement();
6     } // while (rs.next())
7     return false;
8     connection = DriverManager.getConnection
9         (DBServer+databaseName, userName, password);
10     statement.executeUpdate("DELETE FROM Res");
11     while (rs.next()) {
12         ResultSet rs;
13     boolean Emp() {
14         public void init(ServletConfig config) {
15         catch ( Exception e ) {} // Segment 3: start of function Emp()
16         Class.forName(Driver);
17         out.println("<TR><TD> + rs.getString("ResDesc")
18         + "</TD><TR>");
19     void QueryDispTable(HttpServletRequest req, HttpServletResponse res){
20         catch ( Exception e ) {
21         try {
22         } // catch
23         return true;
24         } // emp()
25     } // void QueryDispTable()
26     } // class RsrDB
```

(6 marks)
c) The program *Actors*, based on *Dictionary1* and *RsrDB*, can be regarded as a generalisation of *RsrDB* in the sense that *Actors* involves several lists and associated operations, and each list may be regarded as a generalisation of the concept of an evolving dictionary. Develop this idea as precisely as you can, and draw conclusions of interest to software engineers.

(3 marks)

d) One novel feature illustrated by *Actors* is the concept of *entity referential integrity*. Explain this constraint and give examples drawn from *Actors*.

(3 marks)