

IBM Inter-University Programming Contest 2010 Important Information

Appendix A: Login Information

Appendix B: Point Distribution

Appendix C: How to Submit Answers in Section 1

Appendix D: How to Submit Answers in Section 2

Appendix E: Supplementary Information to Questions 2.1, 2.2,
2.4 and 2.5

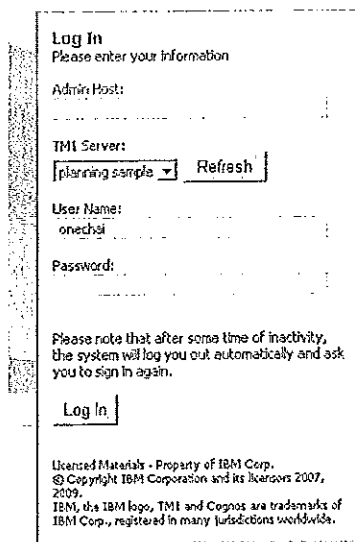
Appendix F: CourseListServlet.java (For Q2.1, Q2.2, Q2.4, Q2.5)

Appendix G: DatabaseAccessUsingSQLJ.sqlj (For Q2.1, Q2.2,
Q2.4, Q2.5)

Appendix C: How to Submit Answers in Section 1

1. You should always use the IBM Cognos server to submit the answers of the multiple choice questions in Section 1.
2. Open an Internet Explorer and type the URL <http://cogserver/tm1web>
Hint: You might use the URL (Section 1 – Multiple Choices) in the Favorites bookmark.

You will see the login page like this,



The screenshot shows a web browser window displaying the login page for the IBM Cognos T1 Web interface. The page has a title bar and a content area. The content area is titled "Log In" and contains the following elements:

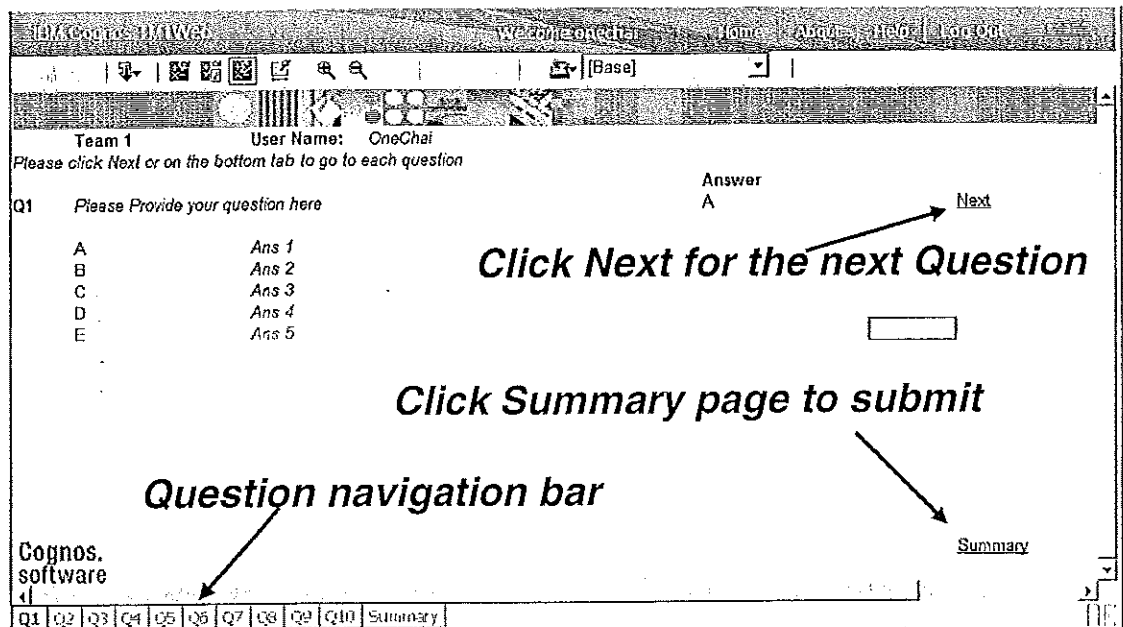
- A heading "Log In" followed by the instruction "Please enter your information".
- A label "Admin Host:" followed by a text input field.
- A label "T1 Server:" followed by a dropdown menu showing "planning sample" and a "Refresh" button.
- A label "User Name:" followed by a text input field containing the text "onechsi".
- A label "Password:" followed by a text input field.
- A paragraph of text: "Please note that after some time of inactivity, the system will log you out automatically and ask you to sign in again."
- A "Log In" button.
- A footer section containing copyright information: "Licensed Materials - Property of IBM Corp. © Copyright IBM Corporation and its licensors 2007, 2009. IBM, the IBM logo, T1 and Cognos are trademarks of IBM Corp., registered in many jurisdictions worldwide."

3. In the Log In screen, just leave the Admin Host blank, and select "planning sample" in the TM1 Server.

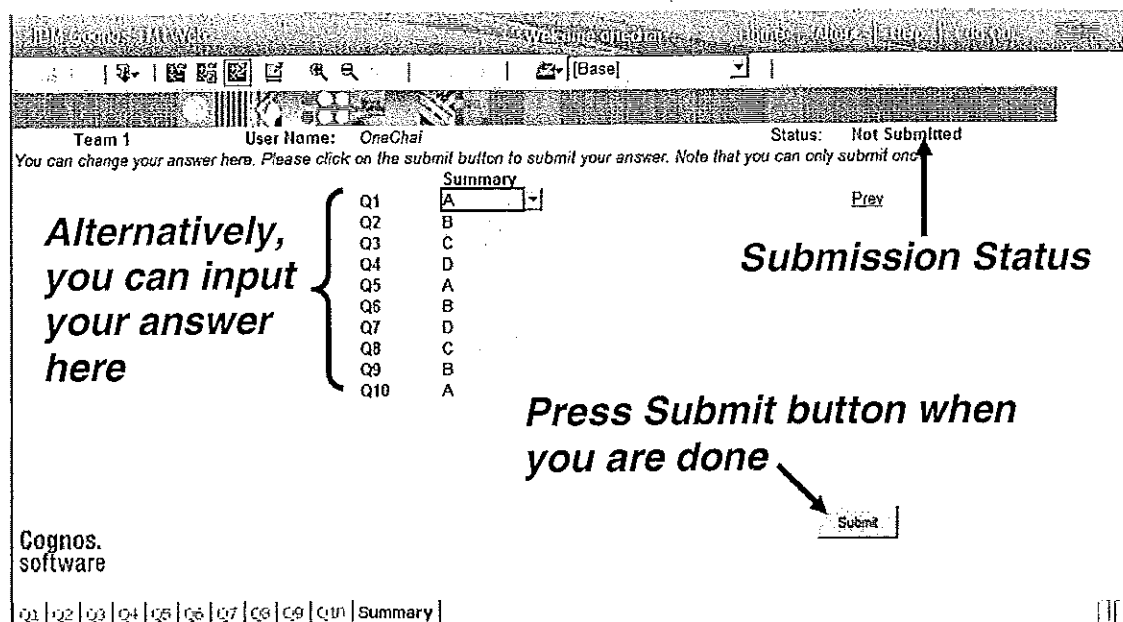
Next, Login into the multiple choice application by the username and password provided in Appendix A. After you logged in, you can check if the team name and user name is correct.

4. After logged in, the first question is displayed and selects the correct answer in the drop down box. Click Next for the next question.

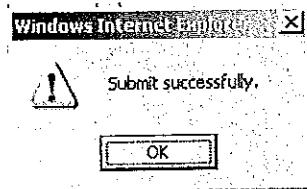
5. To navigate to different questions directly, you may access it through the bottom tab.



6. In order to submit your answer, you must go to the summary page and review your answer. For your convenience, you may also answer the question in this summary page directly from each of the drop down box beside the question number. Note that your Status now should be "Not Submitted". After you confirm it, press the button "Submit".



- Once you submitted, a confirmation box should be appeared. Also check the status should be changed to "Submitted". If you got an error message, please contact your judge.



Team 1 User Name: OneChai Status: Submitted

You can change your answer here. Please click on the submit button to submit your answer. Note that you can only submit once

Summary

Q1	A
Q2	B
Q3	C
Q4	D
Q5	A
Q6	B
Q7	D
Q8	C
Q9	B
Q10	A

Submit

Cognos. software

| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Summary |

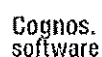
1771

Status get updated

- After you complete the multiple choices, you should send a message with the following format to the judge:

"Team X has completed Section 1. Please evaluate."

The time that you submit this message will be used as the formal time of when you complete the multiple choice section.



If you submit again, you will get error message

10. Once you have submitted your answer, the score will be appeared in the Score Dashboard after a refresh.



11. You can submit all answers in Section 1 once only. And you have to complete all answers together.

Appendix D: How to Submit Answers in Section 2

1. You should always use the IBM Lotus Sametime Connect application as the media to ask questions and submit the answers for questions in Section 2.
2. After you complete a question, you should send a message with the following format to the judge:

"Team X has completed Question 2.1. Please evaluate."

where X is your team number.

3. You may need to provide more information to the judge.
4. After all necessary information is provided to the judge, the judge will return the following message:

"Judge will evaluate Question 2.1 trial Y now."

where Y is the number of trial for this question.

5. The judge will start to evaluate Question 2.1. You can continue to work on the other questions.
6. After a while, the judge will return a message on the points you get for the question:

"You have received 60 points out of 90 points for Question 2.1"

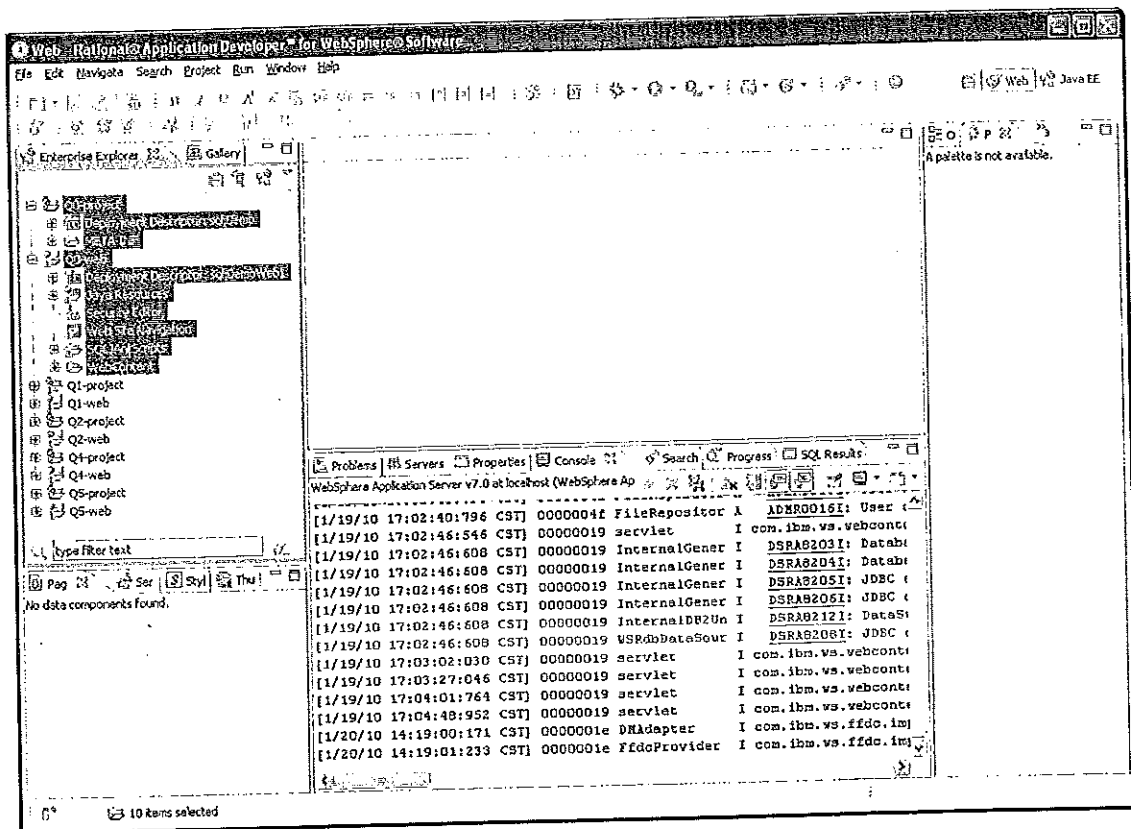
In some cases, the judge will return the reason why points are deduced.

7. If you don't get full mark on the question, you might resubmit the answers for the question again. You have a total of 3 chances to submit the answers for each question in Section 2.

Appendix E: Supplementary Information to Questions 2.1, 2.2, 2.4 and 2.5

This appendix provides important information in order to complete questions 2.1, 2.2, 2.4 and 2.5.

In RAD (i.e., Rational Application Developer), there exists an application called Q0 (i.e., Q0-project and Q0-web) which makes use of the database SAMPLE to produce reports:

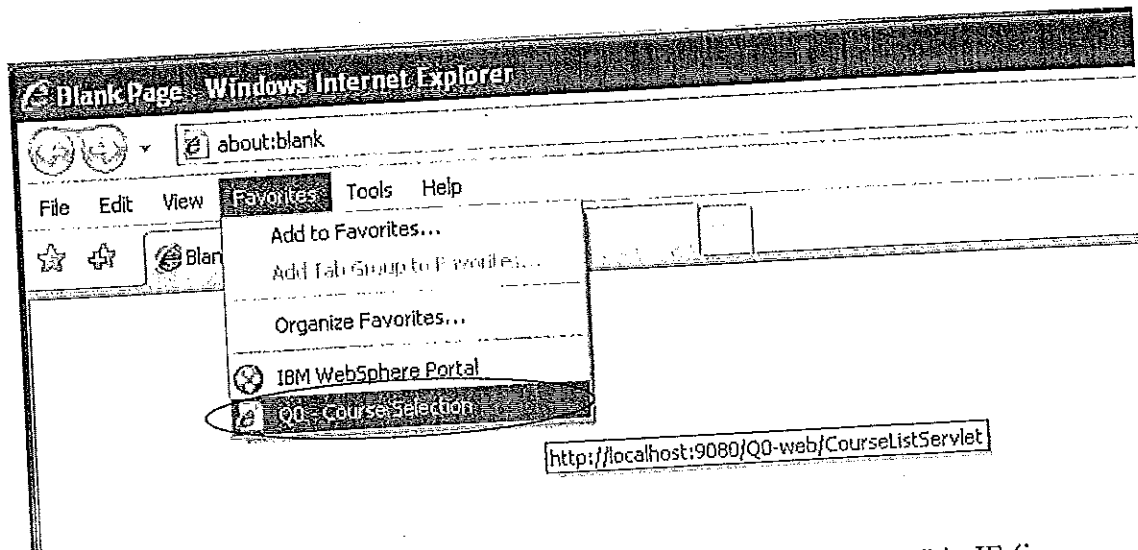


This project Q0 is the same as the project you used in the exercise during the face-to-face training session on Jan 9, 2010.

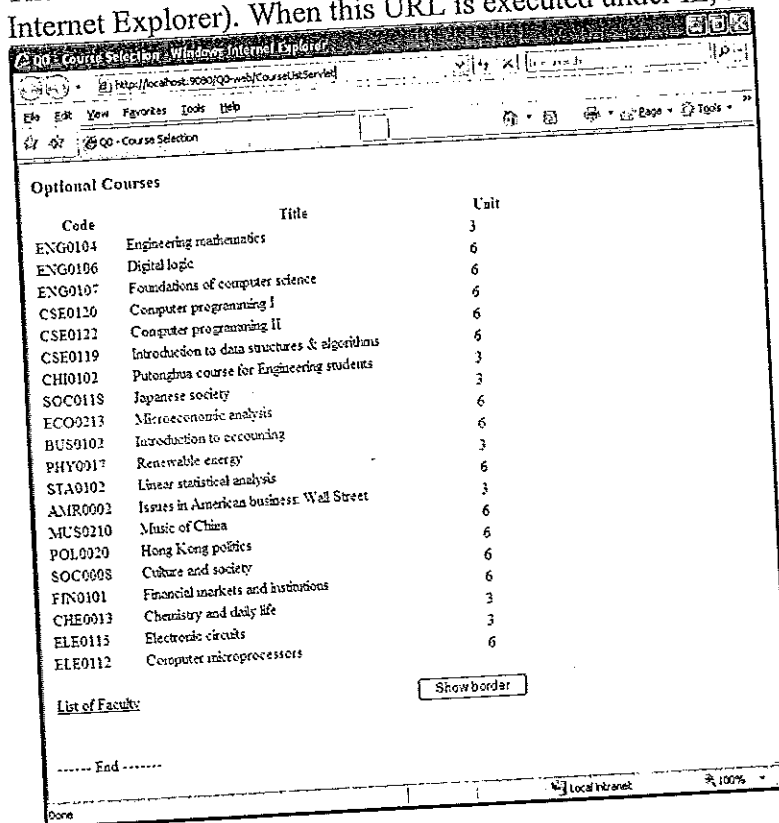
The project Q0 has already been built and deployed to WAS (WebSphere Application Server). This program is triggered by the following URL:

<http://localhost:9080/Q0-web/CourseListServlet>

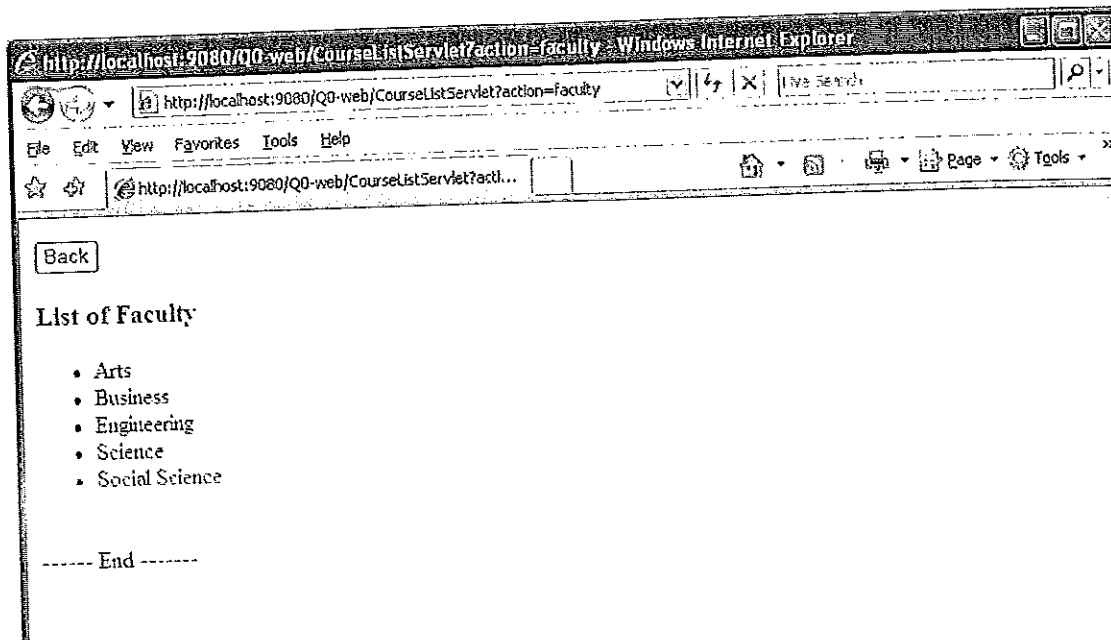
Or you can open Internet Explorer and navigate for the link in Favorites:



The above URL is also bookmarked and available under the "Favorites" in IE (i.e., Internet Explorer). When this URL is executed under IE, the following screen is shown:

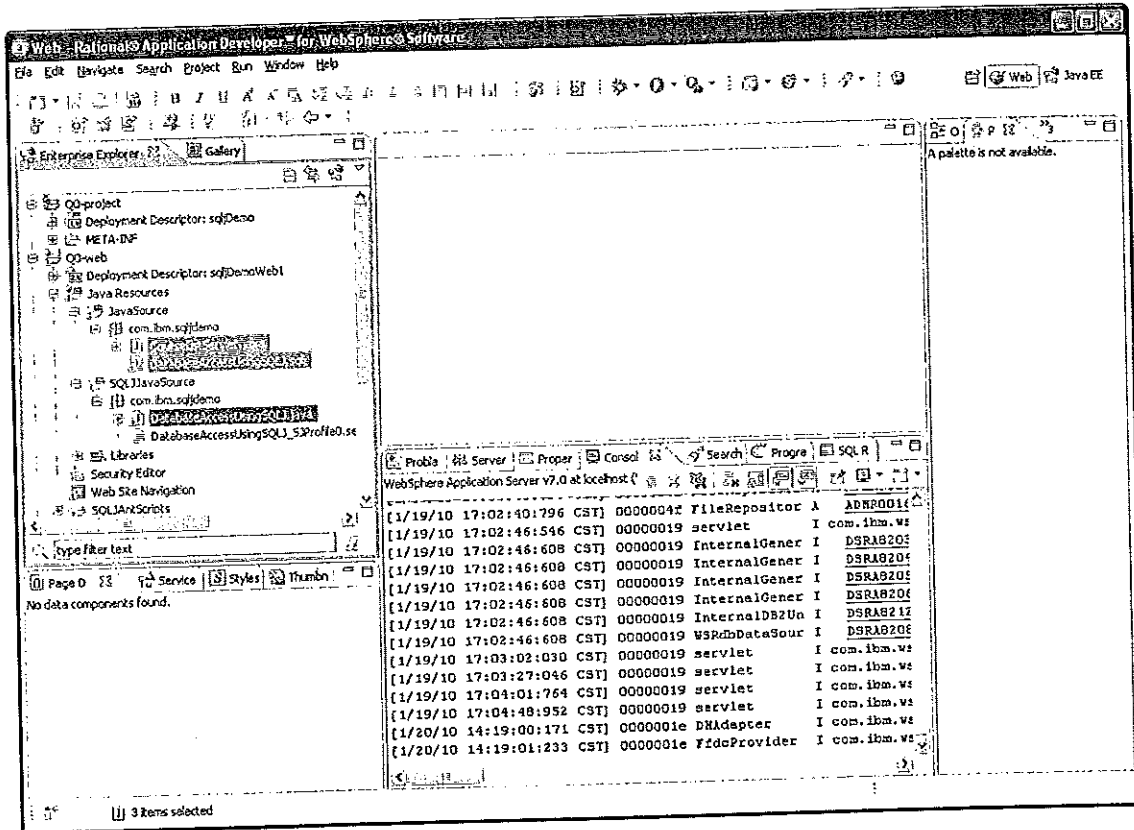


If the link 'List of Faculty' is clicked, the following screen is shown:



If the 'Back' button is pressed, we will return to the first screen.

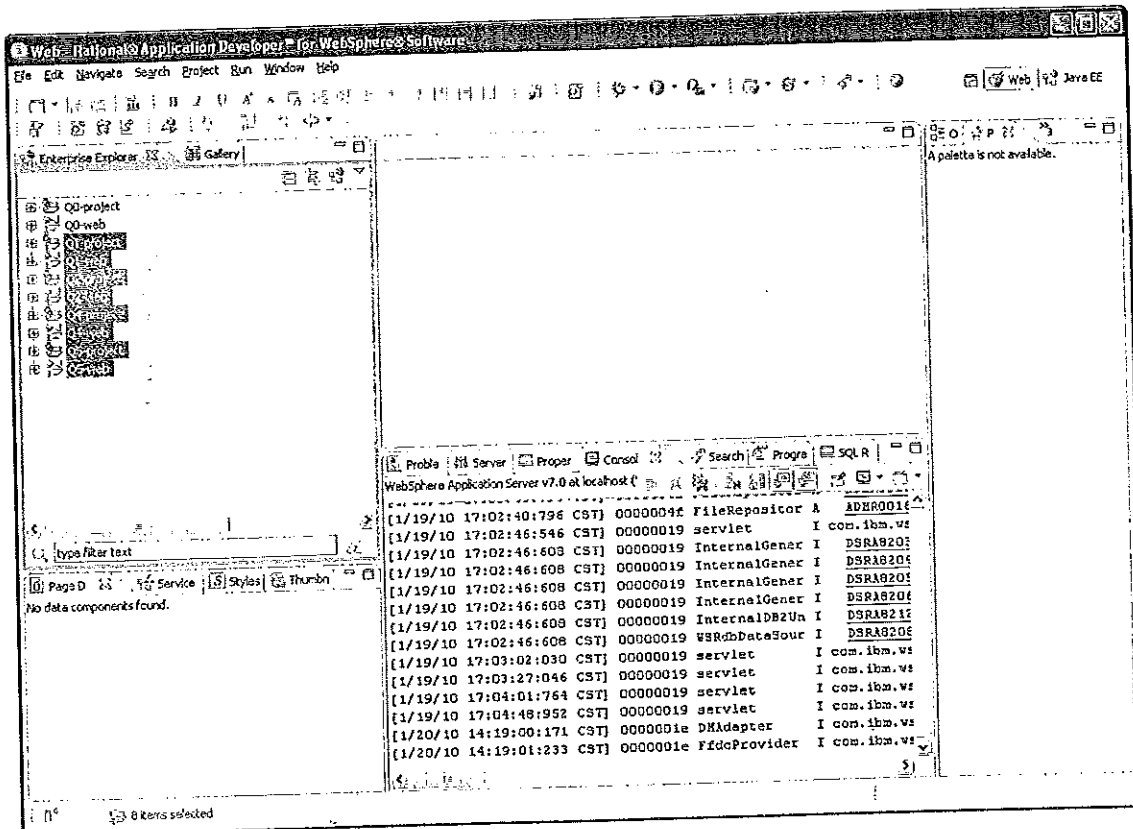
In this application, the code is stored in three programs, namely CourseListServlet.java, DatabaseAccessUsingSQL.sqlj, and DatabaseAccessUsingSQL.java. These programs are stored under the JavaSource in the RAD as shown in the following screen:



The programs make use of the database tables in the database SAMPLE in DB2. The following screen shown how to connect to the database SAMPLE and how to show the list of tables that will be used in this programming contest:

C:\Program Files\IBM\SQLLIB\BIN>db2 list tables			
Table/View	Schena	Type	Creation time
ACT	DB2ADMIN	T	2009-10-20-15.41.01.765009
ADFUSR	DB2ADMIN	S	2009-10-20-15.41.08.218003
CATALOG	DB2ADMIN	T	2009-10-20-15.41.21.671003
CL_SCHED	DB2ADMIN	T	2009-10-20-15.40.56.281002
COURSE	DB2ADMIN	T	2009-12-29-11.37.00.827002
CURRICULUM	DB2ADMIN	T	2009-12-29-11.37.16.515002
CUSTOMER	DB2ADMIN	T	2009-10-20-15.41.20.156003
DEPARTMENT	DB2ADMIN	T	2009-10-20-15.40.56.796003
DEPT	DB2ADMIN	A	2009-10-20-15.40.57.890004
EMP	DB2ADMIN	A	2009-10-20-15.40.58.578004
EMPACT	DB2ADMIN	A	2009-10-20-15.41.01.765000
EMPLOYEE	DB2ADMIN	T	2009-10-20-15.40.57.890008
EMPMDC	DB2ADMIN	T	2009-10-20-15.41.11.968003
EMPPROJECT	DB2ADMIN	T	2009-10-20-15.41.01.578003
EMP_ACT	DB2ADMIN	A	2009-10-20-15.41.01.765005
EMP_PHOTO	DB2ADMIN	A	2009-10-20-15.40.58.578008
EMP_RESUME	DB2ADMIN	T	2009-10-20-15.40.59.328003
FACULTY	DB2ADMIN	T	2009-12-29-11.37.14.686002
INVENTORY	DB2ADMIN	T	2009-10-20-15.41.19.703003
IN_TRAY	DB2ADMIN	T	2009-10-20-15.41.02.187001
ORG	DB2ADMIN	T	2009-10-20-15.41.02.265003
PRODUCT	DB2ADMIN	T	2009-10-20-15.41.18.546003
PRODUCTSUPPLIER	DB2ADMIN	T	2009-10-20-15.41.22.703003
PROJ	DB2ADMIN	A	2009-10-20-15.41.01.203004
PROJECT	DB2ADMIN	T	2009-10-20-15.41.01.203008
PROJECT	DB2ADMIN	T	2009-10-20-15.41.00.218002
PURCHASEORDER	DB2ADMIN	T	2009-10-20-15.41.21.015003
SALES	DB2ADMIN	T	2009-10-20-15.41.02.531003
SCHEDULE	DB2ADMIN	T	2009-12-29-11.37.14.936002
SCHEDULE_REMARK	DB2ADMIN	T	2009-12-29-11.37.15.074002
STAFF	DB2ADMIN	T	2009-10-20-15.41.02.343003
STAFFC	DB2ADMIN	T	2009-10-20-15.41.05.640003
STAFFC	DB2ADMIN	T	2009-12-29-11.37.16.249002
STUDENT	DB2ADMIN	T	2009-12-29-11.37.12.390002
SUBCOURSE	DB2ADMIN	T	2009-12-29-11.37.13.265000
SUBCOURSE_TIME	DB2ADMIN	T	2009-10-20-15.41.22.124003
SUPPLIERS	DB2ADMIN	T	2009-10-20-15.41.02.796003
UACT	DB2ADMIN	U	2009-10-20-15.41.02.859008
UASTRDE1	DB2ADMIN	U	2009-10-20-15.41.02.890003
UASTRDE2	DB2ADMIN	U	2009-10-20-15.41.02.843008
UDEPMG1	DB2ADMIN	U	2009-10-20-15.41.02.624003
UDEPT	DB2ADMIN	U	2009-10-20-15.41.02.749003
UEMP	DB2ADMIN	U	2009-10-20-15.41.02.859003
UEMPDPT1	DB2ADMIN	U	2009-10-20-15.41.03.093003
UEMPLP	DB2ADMIN	U	2009-10-20-15.41.02.843003
UEMPPROJECT	DB2ADMIN	U	2009-10-20-15.41.03.062008
UFORPLA	DB2ADMIN	U	2009-10-20-15.41.02.734003
UHDEPT	DB2ADMIN	U	2009-10-20-15.41.03.078013
UPHONE	DB2ADMIN	U	2009-10-20-15.41.02.765003
UPROJ	DB2ADMIN	U	

In RAD, four other projects (i.e., Q1, Q2, Q4 and Q5) have been created which has the same code as Q0. You might use these four projects to perform the programming work for the questions 2.1, 2.2, 2.4 and 2.5. The following screen highlighted the four project templates:



Appendix F: CourseListServlet.java (For Q2.1, Q2.2, Q2.4, Q2.5)

```

/*(c) Copyright IBM Corp. 2004 All rights reserved.          */
/*                                                             */
/*This sample program is owned by International Business Machines */
/*Corporation or one of its subsidiaries ("IBM") and is copyrighted */
/*and licensed, not sold.                                     */
/*                                                             */
/*You may copy, modify, and distribute this sample program in any */
/*form without payment to IBM, for any purpose including developing,*/
/*using, marketing or distributing programs that include or are */
/*derivative works of the sample program. Licenses to IBM patents */
/*are expressly excluded from this license.                   */
/*The sample program is provided to you on an "AS IS" basis, without */
/*warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL */
/*WARRANTIES EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO*/
/*THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTIC-*/
/*ULAR PURPOSE. Some jurisdictions do not allow for the exclusion or */
/*limitation of implied warranties, so the above limitations or */
/*exclusions may not apply to you. IBM shall not be liable for any */
/*damages you suffer as a result of using, modifying or distributing */
/*the sample program or its derivatives.                     */
/*                                                             */
/*Each copy of any portion of this sample program or any derivative */
/*work, must include a the above copyright notice and disclaimer of */
/*warranty.                                                  */
/*                                                             */
/*****/

//SOURCE FILE NAME: CourseListServlet.java
//SAMPLE: a sample servlet program for generating payroll reports

package com.ibm.sqljdemo;

import java.io.IOException;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.servlet.Servlet;
import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import java.sql.Connection;
import java.sql.SQLException;

```

```

import java.util.*;
import javax.servlet.*;
import javax.sql.*;

/**
 * @version 1.0
 * @author
 */
public class CourseListServlet extends HttpServlet implements Servlet {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    private static DataSource ds = null;

    private final static String courseListPage =
        "<HTML>" + "\n" +
        "<HEAD>" + "\n" +
        " <TITLE> Q0 - Course Selection </TITLE>" + "\n" +
        "</HEAD>" + "\n" +
        "<BODY>";

    /**
     * @see javax.servlet.GenericServlet#void ()
     */
    public void init() throws ServletException {
        super.init();
        InitialContext ctx = null;
        Hashtable parms = new Hashtable();
        parms.put(
            Context.INITIAL_CONTEXT_FACTORY,
            "com.ibm.websphere.naming.WsnInitialContextFactory");
        parms.put(Context.PROVIDER_URL, "iiop:///");

        try {
            ctx = new InitialContext(parms);
            ds = (javax.sql.DataSource) ctx.lookup("jdbc/sqlj_test");

        } catch (Exception e) {
            e.printStackTrace();
        }
    } // end of initialize a database

    /**
     * @see javax.servlet.http.HttpServlet#void
     * (javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
     */

```

```

    public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        doService(req, resp);
    }

    /**
     * @see javax.servlet.http.HttpServlet#void
     * (javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
     */
    public void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        doService(req, resp);
    }

    public void doService(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        ServletOutputStream os;
        os = resp.getOutputStream();

        String tran = req.getParameter("action");
        if (tran == null) {
            displayCourseList(req, resp, os);
        } else if (tran.equals("faculty")){
            displayFacultyList(req, resp, os);
        } else {
            displayCourseList(req, resp, os);
        }
    }

    private void displayCourseList(HttpServletRequest req, HttpServletResponse
resp, ServletOutputStream os){
        try {
            os.println(courseListPage);
            Connection con = ds.getConnection();

            int borderSize = 0;
            String tran = req.getParameter("border");
            if (tran != null) {
                borderSize = Integer.parseInt(tran);
            }

            String courseList = DatabaseAccessUsingSQLJ.getCourseList(con,
os, borderSize);
            os.println(courseList);

            os.println("<BR />");
            os.println("<TABLE WIDTH=\\\"500\\\" BORDER=\\\"0\\\" >");
            os.println(" <TR>");

```

```

        os.println("    <TD VALIGN=\"TOP\">");
        os.println("        <A HREF=\"?action=faculty\">List of
Faculty</A>");
        os.println("    </TD>");
        os.println("    <TD ALIGN=\"RIGHT\" VALIGN=\"TOP\">");
        os.println("        <FORM ACTION=\"CourseListServlet\"
METHOD=\"GET\" >");
        if (tran == null || tran.equalsIgnoreCase("0")) {
            os.println("            <INPUT TYPE=\"hidden\"
NAME=\"border\" VALUE=\"1\" >");
            os.println("            <INPUT TYPE=\"submit\" VALUE=\"Show
border\" >");
        } else {
            os.println("            <INPUT TYPE=\"hidden\"
NAME=\"border\" VALUE=\"0\" >");
            os.println("            <INPUT TYPE=\"submit\" VALUE=\"Hide
border\" >");
        }
        os.println("        </FORM>");
        os.println("    </TD>");
        os.println(" </TR>");
        os.println("</TABLE>");
        os.println("<BR>----- End -----<BR>");
        os.println("</BODY>");
        os.println("</HTML>");
        os.close();
    } catch (IOException ie) {
        System.out.println("IOException:" + ie.toString());
    } catch (SQLException se){
        System.out.println("SQLExcepion:" + se.toString());
    } catch (Exception e){
        System.out.println("Exception:" + e.toString());
    }
}

static void displayFacultyList(HttpServletRequest req, HttpServletResponse
resp, ServletOutputStream os){
    try {
        os.println("<INPUT TYPE=\"button\" VALUE=\"Back\"
onClick=\"history.go(-1)\" >");
        os.println("<BR />");
        os.println("<H3>List of Faculty</H3>");
        os.println("<UL>");
        os.println("    <LI>Arts");
        os.println("    <LI>Business");
        os.println("    <LI>Engineering");
        os.println("    <LI>Science");
        os.println("    <LI>Social Science");
        os.println("</UL>");
    }
}

```

```
        os.println("<BR>----- End -----<BR>");
        os.println("</BODY>");
        os.println("</HTML>");
        os.close();
    } catch (IOException ie) {
        System.out.println("IOException:" + ie.toString());
    } catch (Exception e){
        System.out.println("Exception:" + e.toString());
    }
}

    public void doMenu(HttpServletRequest req, HttpServletResponse res,
ServletOutputStream os)
        throws ServletException, IOException
    {
        } // End of doMenu
}
```

Appendix G: DatabaseAccessUsingSQLJ.sqlj (For Q2.1, Q2.2, Q2.4, Q2.5)

```

/*(c) Copyright IBM Corp. 2004 All rights reserved.          */
/*                                                             */
/*This sample program is owned by International Business Machines */
/*Corporation or one of its subsidiaries ("IBM") and is copyrighted */
/*and licensed, not sold.                                       */
/*                                                             */
/*You may copy, modify, and distribute this sample program in any */
/*form without payment to IBM, for any purpose including developing,*/
/*using, marketing or distributing programs that include or are */
/*derivative works of the sample program. Licenses to IBM patents */
/*are expressly excluded from this license.                     */
/*The sample program is provided to you on an "AS IS" basis, without */
/*warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL */
/*WARRANTIES EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO*/
/*THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTIC-*/
/*ULAR PURPOSE. Some jurisdictions do not allow for the exclusion or */
/*limitation of implied warranties, so the above limitations or */
/*exclusions may not apply to you. IBM shall not be liable for any */
/*damages you suffer as a result of using, modifying or distributing */
/*the sample program or its derivatives.                         */
/*                                                             */
/*Each copy of any portion of this sample program or any derivative */
/*work, must include a the above copyright notice and disclaimer of */
/*warranty.                                                      */
/*                                                             */
/*****/

//SOURCE FILE NAME: DatabaseAccessUsingSQLJ.sqlj
//SAMPLE: the sqlj program for generating payroll reports
//
//SQL Statements USED:
//      SELECT
//
//*****
//
//For more information on the sample programs, see the README file.
//
//For information on developing SQLJ applications, see the Application
//Development Guide.
//
//For information on using SQL statements, see the SQL Reference.
//
//For the latest information on programming, compiling, and running DB2
//applications, visit the DB2 application development website at
// http://www.software.ibm.com/data/db2/udb/ad

```

```

//*****

package com.ibm.sqljdemo;

import java.sql.*;
import sqlj.runtime.ref.*;
import javax.servlet.*;
import java.util.Vector;

#sql iterator CourseInfo(String, String, String);
#sql context ctx;

public class DatabaseAccessUsingSQLJ {

    public static String getCourseList(Connection con, ServletOutputStream os, int
borderSize) {
        StringBuffer courseListPage = new StringBuffer("ABC");

        CourseInfo results;
        String code = null;
        String title = null;
        String unit = null;
        try {
            ctx thisCTX = new ctx(con);
            courseListPage.setLength(0);
            courseListPage.append("<H3> Optional Courses </H3>" + "\n");
            courseListPage.append("<TABLE WIDTH=\"500\" BORDER=\"" +
borderSize + "\" >" + "\n");
            courseListPage.append(" <TR>" + "\n");
            courseListPage.append(" <TH>Code</TH>" + "\n");
            courseListPage.append(" <TH>Title</TH>" + "\n");
            courseListPage.append(" <TH>Unit</TH>" + "\n");
            courseListPage.append(" </TR>" + "\n");
            #sql [thisCTX] results={SELECT code, title, unit FROM COURSE};

            while (true) {
                #sql {FETCH :results INTO :code, :title, :unit};
                if (results.endFetch()) {
                    break;
                }
                courseListPage.append(" <TR>" + "\n");
                courseListPage.append(" <TD>" + code + "</TD>" +
"\n");
                courseListPage.append(" <TD>" + title + "</TD>" +
"\n");
                courseListPage.append(" <TD>" + unit + "</TD>" +
"\n");
                courseListPage.append(" </TR>" + "\n");
            }
        }
    }
}

```

```
        courseListPage.append("</TABLE>");
        results.close();
    } catch (Exception e) {
        System.out.println("Exception:" + e.toString());
    }
    return courseListPage.toString();
}
}
```

