Course Evaluation System Documentation

Jonathan Hoult
jon@coe.drexel.edu

Drexel University
Electrical and Computer Engineering Department
32nd and Chestnut Streets
Philadelphia, PA 19104

http://www.drexel.edu

Gateway Engineering Education Coalition

http://www.gatewaycoalition.org

This document describes a web based software package that can be used to survey students about course related issues. It also contains step-by-step instructions and hints on how to customize it for other surveys. This work has been supported in part by the National Science Foundation, Awards EEC-9109794 and EEC-9727413. Appreciation is also expressed to Polytechnic University of Brooklyn, NY for providing their original contribution to this project as part of the Gateway Engineering Education Coalition.

Software Overview

The Course Evaluation software is available for download at http://www.gatewaycoalition.org and was designed at Polytechnic University of Brooklyn and then modified at Drexel University to work with the local system. Features were added based on feedback from students and the College of Engineering Assessment Committee. In it's current state, it allows for students to complete course evaluations, a special ABET A-K objectives survey, Faculty responses to those surveys, and a Senior Exit survey using the Internet and a web browser. One other type of survey is the Humanities survey that spans multiple classes and terms. Other types of surveys are possible using this software with minimal or no changes to the software structure.

Students receive a customized email containing a list of courses available to evaluate and a web site address for the survey. Supplemental surveys are added to the student's list of courses. Students complete a special ABET A-K survey twice a year. We are not currently using the Senior exit survey, due to it being run on another server. The framework is already set up and configured, though it is currently disabled.

Students connect to the web site and login using their first and last names as well as their birth date. If this method fails due to incorrect information in the database, they can also log in using their student ID number. After they are authenticated, they view a list of courses to evaluate plus any of the additional surveys. Each of the courses has a set of objectives that students use to self-evaluate their competencies from "No Experience" to "Expert". A section is included in addition to the course objectives that allows the students to evaluate the course and the instructor with some of the ABET A through K principles in mind. Fields are available that allow students to enter comments. At Drexel University, a student takes an average of 2 minutes to complete a survey for one course.

The results are anonymously stored in a database for analysis. The numeric averages for the course objectives are displayed on the web and are made available at the close of the term surveys. An access control mechanism allows for the results to be displayed only on campus computers. The full course reports with numeric averages and course comments are individually emailed to the faculty and the department heads. Further analysis is done offline through Microsoft Access. Real-time statistics regarding the number of responses is also displayed on the web during the survey process to allow faculty to identify classes that may not be participating in the process.

The first run with the Course Evaluation system may take one or two weeks for the initial setup by an experienced CGI programmer. Subsequent surveys require approximately 30 minutes to setup. Administrative time and setup time is reduced by automation through the use of scripts. A personalized email is sent to the faculty displaying the courses they are teaching and announcing the start and stop dates of the course evaluation. Students receive weekly reminder emails containing a list of courses that they have not surveyed through the use of a script. After the survey period is over, faculty are emailed the results by a script and more detailed results are put online. The ABET A through K reports are also placed online.

Hardware and Software Requirements

The software was developed and tested on Sun Solaris and Linux systems at Polytechnic University and Drexel University. The disk and memory requirements vary depending on the number of students in the system. A disk with 100 Mbytes of free space should be enough to fit the software and several terms of results. Drexel has run 18 terms worth of surveys, and the data is taking approximately 70MB of space. Memory requirements vary per platform. Drexel uses a SunFire V880 system with 8Gbytes of RAM for the surveys. A PC with a 400 MHz Pentium should work but may be slow for universities with many students.

The Course Evaluation System relies on open source software that is available on Internet and free of charge. Binaries are available for a variety of platforms, but not all. A C compiler is required to build binaries if needed. These packages are also available for Windows NT, but have not been tested and will require a resident expert to implement. It should work with little modification to the Course Evaluation software but would require effort to get the support packages together.

MySQL 3.22.32

MySQL is a database application that uses SQL syntax. It has OODBC support so data can be linked across the network to other databases. Configuration files are available on the MySQL web site that allows Microsoft Access to download data from the server. MySQL and documentation is available at http://www.mysql.com.

Perl 5.005

Perl is a scripting language that is commonly used to write CGI programs for the Web. It is used for other non-Web applications as well. The survey software and support scripts are written in Perl. Perl comes with most Linux distributions. Sun Solaris and other commercial UNIX users install it. Perl is available at http://www.perl.org.

DBI 1.13

The DBI package is an add-on module that fits into Perl. It allows Perl to issue commands to the MySQL databases. It is available at http://www.perl.com/CPAN.

Apache 1.3.27

Feedback from students has indicated that they want to enter their personal information for authentication using a secure socket web server. Drexel uses Apache and Apache with SSL. Apache is available at http://www.apache.org. The SSL patch allows Apache to operate as a secure server. The patches are available at http://www.apache-ssl.org and may be subject to government restrictions. The web site has more details. Other versions of Apache and other web server packages will work as well.

Sendmail

The Perl scripts rely on Sendmail for mail delivery. Most UNIX systems already have Sendmail installed or an equivalent mail transport agent. Sendmail is available at http://www.sendmail.org.

Installation of Course Evaluation System

The following steps should be followed to install the Course Evaluations System. The software is available on the web at http://www.gatewaycoalition.org. The support applications are a pre-requisite for the installation of the Course Evaluation System. For instructions on installation of the support packages, please use the documentation that came with them.

Step 1: Unpacking the Files

After acquiring the evaluation software, copy it to a directory that is accessible on the web like /htddocs or the equivalent. Extract the files from the archive using gzip and tar. WinZip can be used for Windows systems.

Example:

```
     csh% gzip -dc course-eval.tar.gz | tar -xvf -
```

The table below shows the files that will be extracted with a very brief description of the most important ones.

eval-1.1/images/Expert.GIF	Image used by survey
eval-1.1/images/Fair.GIF	Image used by survey
eval-1.1/images/Good.GIF	Image used by survey
eval-1.1/images/No_Exp.GIF	Image used by survey
eval-1.1/images/No_Resp.GIF	Image used by survey
eval-1.1/images/Poor.GIF	Image used by survey
eval-1.1/help.html	An empty helpfile
eval-1.1/scripts/build_enrollment.pl	Helps stats_class.cgi
eval-1.1/scripts/create.tables.sql	Creates SQL tables
eval-1.1/scripts/extract.pl	Loads data into tables
eval-1.1/scripts/faculty-mail.pl	Sends notices to faculty
eval-1.1/scripts/nosurvey.sql	Removes certain classes from survey
eval-1.1/scripts/send-mail-students.pl	Sends notices to students
eval-1.1/thankyou.html	Thank you page.
eval-1.1/login2.html	Alternative login by student ID
eval-1.1/templates/stats_class.html	Class-by-Class statistics
eval-1.1/templates/template_ABET.html	Special ABET Survey
eval-1.1/templates/profresults_template_core.html	Faculty Report for courses with objectives
eval-1.1/templates/profresults_template_nocore.html	Faculty Report for courses without objectives
eval-1.1/templates/CE_survey00.html	Senior Survey
eval-1.1/templates/CHE_survey00.html	Senior Survey
eval-1.1/templates/CIVE_survey00.html	Senior Survey
eval-1.1/templates/MATE_survey00.html	Senior Survey
eval-1.1/templates/AE_survey00.html	Senior Survey

eval-1.1/templates/MEM_survey00.html	Senior Survey
eval-1.1/templates/EE_survey00.html	Senior Survey
eval-1.1/templates/template_core.html	Survey for courses with objectives
eval-1.1/templates/template_nocore.html	Survey for courses without objectives
eval-1.1/templates/results_template_core.html	Report for courses with objectives
eval-1.1/templates/results_template_nocore.html	Report for courses without objectives
eval-1.1/templates/objectives_template.html	Report for objectives
eval-1.0/cgi-bin/quest.cgi	Presents survey to students
eval-1.1/cgi-bin/user.cgi	Login and survey list
eval-1.1/cgi-bin/pubresults.cgi	Displays numeric results (public information)
eval-1.1/cgi-bin/objectives.cgi	Displays objectives for all courses
eval-1.1/cgi-bin/objectives_display.cgi	Displays objectives for a specific course
eval-1.1/cgi-bin/profresults.cgi	Displays numeric results and comments
eval-1.1/cgi-bin/profresults_process.cgi	
eval-1.1/cgi-bin/stats_class.cgi	Displays class-by-class response rates
eval-1.1/cgi-bin/pubresults_process.cgi	Displays numeric results (public information) for a specific course
eval-1.1/index.html	Entry point
eval-1.1/pubresults.html	Template for public results
eval-1.1/noaccess.html	Access denied page
eval-1.0/quest.conf	Configuration file
eval-1.1/objectives.html	Template for objectives.cgi
eval-1.1/profresults.html	Template for profresults.cgi

Step 2: Creating the Databases

An empty database can be created with the `mysqladmin` command as the MySQL administrator.
Example:

```
csh% mysqladmin create survey2000sum
```

Currently five tables are used to hold the information required. The Answers table holds the responses as entered by the students. The CourseList table holds the class listings for all of the students. There is one entry in this table for every class a student is enrolled. The Courses table holds information about the course including instructor, course title, email address of the instructor and other information. The Questions table holds objectives for each of the courses. The Students table holds demographic information for the students as well as authentication information like their birth date and Student ID number.

The tables can be created using the `create.tables.sql` script creates the tables and the fields required by the software. The first line of the script should be changed to reflect the database used.

Example:

```
     csh% mysql < create.tables.sql
```

Create a username and assign access permissions to these tables in MySQL. This is not necessarily on at the system level in /etc/passwd but one local to the MySQL application

Example:

```
     mysql> GRANT  SELECT, UPDATE, INSERT ON survey2000sum.*
     TO eval@localhost IDENTIFIED BY ‘secret_password’
```

Step 3: Loading the Databases

Drexel University uses Banner software for administrative record keeping. A form was created by the Banner support group that gives users the ability to download student records onto a PC. The comma-delimited file containing the student information is transferred to the server using E-mail or FTP from the PC. The extract.pl script populates the tables in the MySQL database.

Example:

```
     csh% extract.pl student_records.csv
```

The extract script has been used on data that was generated by administrative systems other than banner with some changes. It is highly probable that this script will need modifications to work with other administrative systems. Most of the time required for setup is this step.

```
     csh% FRF-data-import.pl
```

This step fills in the required fields for the Faculty response form. It pulls data from the courses table to determine what professors are teaching, and puts them in a list similar to CourseLists for the students for their Faculty response forms.

Inevitably, some courses in the database may not be surveyed for a variety of reasons. Two fields are in place that allows the administrator to enable or disable courses for survey. By setting CourseList.Prompt to ‘0’ it is removed from a student’s list of courses to evaluate. By setting Courses.Prompt to ‘0’, it is removed from the list of all of the students in the class. By setting these values to ‘1’, it enables a course of evaluation.

To disable a course for one student:

```
     mysql>   UPDATE   CourseList   SET   Prompt='0'   WHERE
ID='123456789';
```

To disable a course for the entire class:

```
     mysql> UPDATE Courses Set Prompt='0' Where CRN='12345';
```

Step 4: Installing the Configuration File

Move the quest.conf file to /var/web/quest.conf. As root, set the owner of the configuration file to the same user id as the web server process. Set the permissions so that only the web server process can read it. The configuration file has one entry per line. The first line should contain the MySQL user set in Step 2. The second line should be the password. The third line should be set to the top level directory of the course evaluation software. The forth line contains background color and is only used by some portions of the software. The fifth line contains the name of the database to use for surveying and reporting.

Example:

```
     csh% chown web /var/web/quest.conf
     csh% chmod 600 /var/web/quest.conf
```

Step 5: Installing Course Objectives

Drexel University has objectives that are identified for each course. The students are evaluated on these objectives when surveying courses. The objectives are stored in the Questions database. They are put into a comma-separated file with Access and sent as an email to the server. They are loaded into the database with the SQL load command. The file will most likely need to be converted to a Unix text file before loading into the database.

Example:

```
     csh% dos2unix questions.csv questions.txt
     mysql> load data infile "/tmp/questions.txt" into table
     Questions fields terminated by ",";
```

The survey can operate without objectives for some or all of the courses. If course objectives are not present, a shortened survey is given that only has questions 13 through 24 and the comment boxes as shown in the samples at the send of this document.

Step 6: Installing Customized HTML files

Many of the HTML files the students see contain graphics and information that are specific to Drexel University. Modification is required to the existing HTML files to fit styles and content appropriate for other universities. Some of the HTML is embedded in the CGI scripts and will require modification. These will be removed from the CGI scripts in later versions of the Course Evaluation software to make the system more portable.

ABET and Senior Surveys are provided. They are surveys that can be added to the list of course surveys that students see. Adding HTML files with different input fields can create other surveys.

Step 7: Activating the Report System

It is useful to have the class-by-class statistics reporting system active before the system "goes live". The `statsclass.cgi` script can display the number of responses for classes and departments. It also calculates the overall responses. We encourage the students to participate in a variety of ways. The report allows the students and the faculty to see the number of responses. Since it shows responses by department, an inter-department contest can be announced to promote the use of the system. The build_enrollment.pl script was written to make this reporting appear very fast. It should be run every 15 or 30 minutes though the UNIX cron daemon.

The `pubresults.cgi` show the numeric averages for each individual class. If results are to be made public, the script should be updated at the end of the term's survey to give access to the appropriate database. This prevents anybody from seeing the results before the survey is over and grades are submitted. The `profresults.cgi` script shows the numeric results and the comments submitted by the students. This program limits access to one or two hosts defined in the script. This could be limited to administrative stations for examining the data while the survey is run. The `profresults.cgi` script is also used to generate final reports to the faculty.

The objectives.cgi script shows the course objectives for all of the courses on the database. Access to the objectives for all of the courses is a benefit to the students and helps satisfy some of the ABET requirements.

Step 8: Notifying Faculty and Students

After the system has been tested and running, notifications should be sent to faculty and students. At Drexel, faculty receive an email indicating which courses will be surveyed and when they will be surveyed. Faculty usually submit corrections and other requests if the administrative information obtained in step three is incorrect.

After the faculty have been notified and corrections have been made, students should be notified that the survey has started and when it will stop. At Drexel, we set the web browsers in public labs to point to the course evaluation page. We post signs in highly visible areas such as departmental offices, elevators, restroom doors and labs.

Our experience has shown that the customized email is the most effective. We send a customized email to the students notifying them that the course evaluations are available and display the courses they are taking which require an evaluation. We send weekly reminders to the students that show the courses that have not been evaluated. Students who have completed the evaluations do not get the reminder email. Students seem to respond better when the email is addressed them by their first name at the beginning of the message. Most of the responses are received immediately following the email. The email also includes a link they can click on so that the users will not have to log in if they use the link. This also saves them some time, and makes them more willing to do the surveys.

The script that sends the notices to the faculty is `faculty-mail.pl`. The script that sends the notices to the students is `send-mail-students.pl`. The messages that are sent in these scripts should be customized to adjust to the needs of other institutions.

Step 9: Reporting

The `stat_class.cgi` script shows real-time statistics of the responses per department and class. This is used so that faculty can judge how many have responded and how much reminding the students might need. It is also used to create competition between departments.

The `objectives.cgi` script shows the objectives identified with each course. This is useful for faculty to display the information that will be surveyed. This may also help satisfy some of the ABET requirements. Students can use it to get a better idea of what a course is about before they take it.

The `pubresults.cgi` and `pubresults_process.cgi` are the most popular. These are updated at the end of each survey and allow the students to view the averages of the numerical portion of the survey. These pages are only available to computers in the Drexel.edu domain. This portion of the code will be changed in future versions to make it more portable for other institutions.

The `profresults.cgi` and `profresults_process.cgi` scripts display the numerical averages and comments for a course. Access is limited to a select hostnames. This script is also used to email results to the faculty.

Other reports are possible through Microsoft Access or other database packages. MySQL ODBC drivers are available at the MySQL web site for Windows 95/98/NT and 2000.

Miscellaneous Notes

Adding A Custom Survey

Other surveys can be added that do not follow the form of the course survey. Drexel University has Senior Surveys and special ABET surveys that fall into the category of other. We also do a faculty response form, as well as a Humanities survey. For other surveys, an HTML template should be created using any of the available HTML authoring tools and placed in the template directory in the course evaluation directory. Fields that return a single input value should use input field names A1 though A12, B1 though B12. Comments boxes should use C1 through C12. Records should be created for each of the students taking the survey in the CourseList table. Logic will need minor changes in the quest.cgi script to prompt the students with the correct template. This will include adding a custom CRN number to identify the survey. We are currently using CRNs between 99990 and 99999 to identify custom surveys.

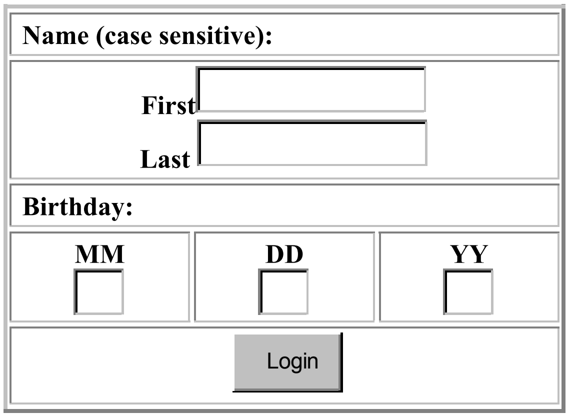
Other Contributions

We are interested in comments, suggestions and new features added to this software. If you have a way to make this software or document better, please let us know.

Sample Screens From Web

Sample Login Screen

The following shows a student logging into the Course Evaluation System. After login, the Course Selection Screen in displayed.



Name (case sensitive):

First ____

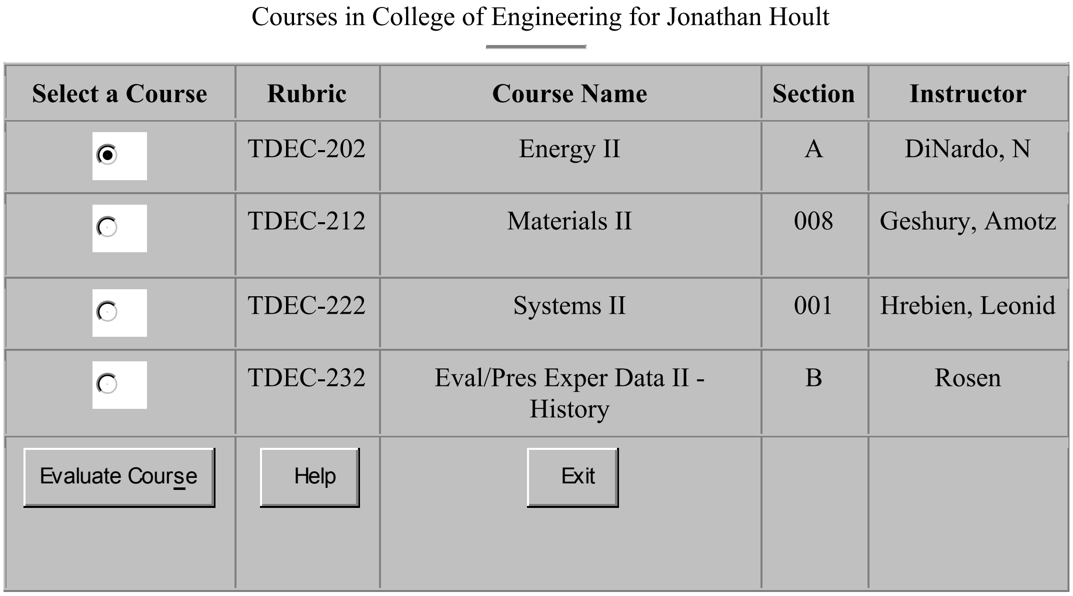
Last ____

Birthday:

MM	**DD**	**YY**
____	____	____

Sample Course Selection Screen

The following shows a list of courses that a student can evaluate. After selecting a course, the Course Survey Screen appears.

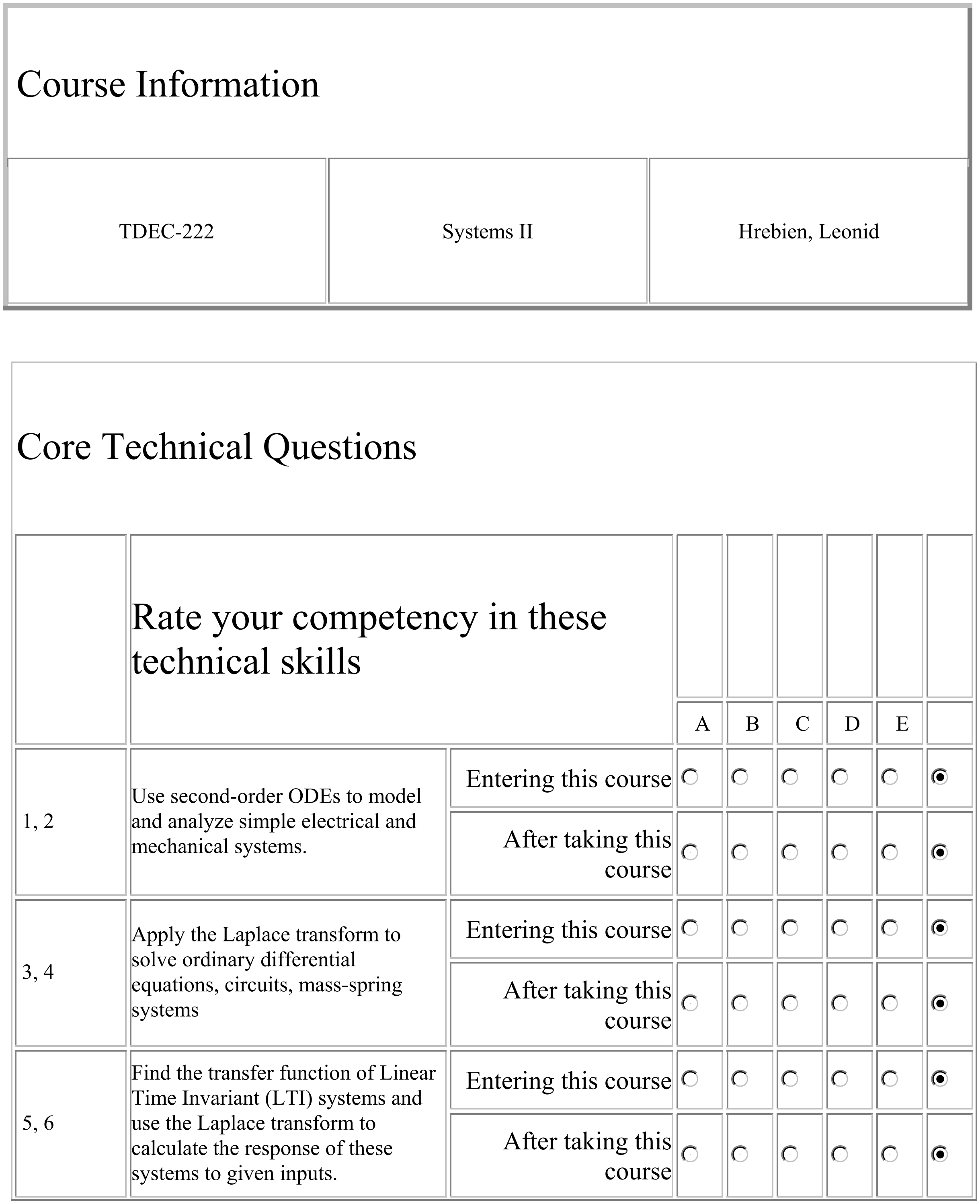


Courses in College of Engineering for Jonathan Hoult

Select a Course	Rubric	Course Name	Section	Instructor
●	TDEC-202	Energy II	A	DiNardo, N
○	TDEC-212	Materials II	008	Geshury, Amotz
○	TDEC-222	Systems II	001	Hrebien, Leonid
○	TDEC-232	Eval/Pres Exper Data II - History	B	Rosen
Evaluate Course	Help	Exit		

Sample Course Survey With Course Objectives

The Course Survey Screen contains the questions for a course. Alternative survey screen can appear for courses without objective questions, special ABET surveys or Exiting Senior Surveys. Other forms can be added.



Course Information

TDEC-222	Systems II	Hrebien, Leonid

Core Technical Questions

	Rate your competency in these technical skills		A	B	C	D	E	
1, 2	Use second-order ODEs to model and analyze simple electrical and mechanical systems.	Entering this course	○	○	○	○	○	●
		After taking this course	○	○	○	○	○	●
3, 4	Apply the Laplace transform to solve ordinary differential equations, circuits, mass-spring systems	Entering this course	○	○	○	○	○	●
		After taking this course	○	○	○	○	○	●
5, 6	Find the transfer function of Linear Time Invariant (LTI) systems and use the Laplace transform to calculate the response of these systems to given inputs.	Entering this course	○	○	○	○	○	●
		After taking this course	○	○	○	○	○	●

7, 8	Manipulate simple continuous-time and discrete-time signals, find odd and even parts and classify these-time systems based on linearity, time-invariance, causality, use of memory, and stability.	Entering this course	○	○	○	○	○	●
		After taking this course	○	○	○	○	○	●
9, 10	Manipulate the Dirac and Kronecker delta functions.	Entering this course	○	○	○	○	○	●
		After taking this course	○	○	○	○	○	●
11, 12	Use the impulse response to calculate the response of LTI systems to given inputs, perform convolution in the time domain for continuous-time and discrete-time systems.	Entering this course	○	○	○	○	○	●
		After taking this course	○	○	○	○	○	●

Course and Instructor Questions

Please respond to the following statements about this course:		To a Very Great Extent	To a Great Extent	To a Moderate Extent	To a Limited Extent	Not at All	No Selection
		A	B	C	D	E	
13	Teamwork or group projects and assignments were an integral part of this course	○	○	○	○	○	●
14	The course had a multi-disciplinary perspective	○	○	○	○	○	●
15	Computer technologies were used to enhance my learning.	○	○	○	○	○	●

16	The course objectives and requirements were clearly communicated	○	○	○	○	○	●
17	What is your overall rating of the course (A = outstanding, C = average, E = poor)	○	○	○	○	○	●
18	What is your rating of the textbook used in this course? Your textbook comments can be typed below. (A = outstanding, C = average, E = poor)	○	○	○	○	○	●
Please respond to the following statements about this course:		To a Very Great Extent	To a Great Extent	To a Moderate Extent	To a Limited Extent	Not at All	No Selection
		A	B	C	D	E	
19	The instructor was well prepared for the lectures	○	○	○	○	○	●
20	The instructor’s communication skills were good	○	○	○	○	○	●
21	The instructor’s attitude toward the students was positive and helpful	○	○	○	○	○	●
22	The instructor provided timely feedback on student performance	○	○	○	○	○	●

23	What is your overall rating of the instructor? (A = outstanding, C = average, E = poor)	○	○	○	○	○	●
24	What grade do you expect to receive from this class?	○	○	○	○	○	●

List the key strengths and weaknesses of this course.

List the key strengths and weaknesses of the instructor.

Enter other comments that you might have about this course here.

.