OOD130I

Project Number: SYS 9911-51

# Development of Interactive Web-based Educational Modules

Interactive Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Yung Giang

Yung Giang

Date: May 1, 2000

Approved:

[illegible]

Professor Satya Shivkumar

## Table of Contents

## List of Illustrations

## Abstract

The purpose of this project was to develop interactive web-based educational modules. Several modules were developed in HTML, VRML, Java, and Perl for an introductory course in material science. The educational modules can be used to improved student – faculty communication and to enhance active student learning beyond the classroom.

# 1.0 Introduction

The Internet is a fairly new medium used to exchange ideas. Over the past several years, the Internet has grown more and more popular. The explosive use of the Internet is both beneficial and detrimental to society.

The Internet can have many educational uses as well, especially in universities and colleges. Practically all universities and colleges have web sites, where they promote the school and the available programs. Many universities also have on-line applications which allow potential students to apply to the school without using paper forms. The Internet has even filtered its way into some high schools and elementary schools where it is being used to post announcements and events. Remote learning is also an important aspect of using the Internet for educational purposes. Students can take courses using the Internet. This is especially helpful for those who have inconveniences, which prevent them from attending the regular lectures for the course. Examples of people with inconveniences include those who have to work and those with a physical disability. The Internet is another channel in which students may interact with professors and with other students.

The purpose of this project was to create interactive web-based educational modules for a course taught at the Worcester Polytechnic Institute. These interactive web-based modules are important for improvement of communication between the professor and students. These modules also act as a supplement to lecture to ensure that the students have enough resources when attempting to learn the material. In addition, these modules are another media in which the professors can obtain feedback from students on a particular subject.

## 2.0 Objective

The main objectives of this project were:

- To evaluate the use of Internet at various institutions.
- To examine the capabilities of the internet to improve interactive learning and to enhance student - student communication and student - instructor communication
- To develop an interactive web-based graphical interpretation of simple engineering equations
- To develop course-specific educational modules

# 3.0 Use of Internet in Education

In this section, previous, recent and current learning methods are discussed so that there is a basis for the research in this project.

## 3.1 Traditional Methods of Learning

For many years, learning consisted of lectures, homework, and exams. All the teaching is basically done in the classroom by professors during lectures. Students must attend the learning forum to gain any valuable information.

This traditional way of learning is extremely inflexible. In the instance of a typical college course, there is very little interaction between professors and students except for those times designated as lecture and office hours. Conflicting schedules often prevent a student from attending office hours to seek help on a particular subject.

## 3.2 Recent Developments in Education Modules

There are many schools in the United stated that have begun using distance learning technology. Some of the Universities and colleges, including UMASS, WPI, and State University of New York, have made or purchased their interactive systems. The popular systems that I have researched are WebCT, ILN (interactive learning network, Black board, TopClass, CyberProf, and ReCourse.

Interactive web based modules are in use at a number of major schools. Some of the schools in which this software is being used are listed below.

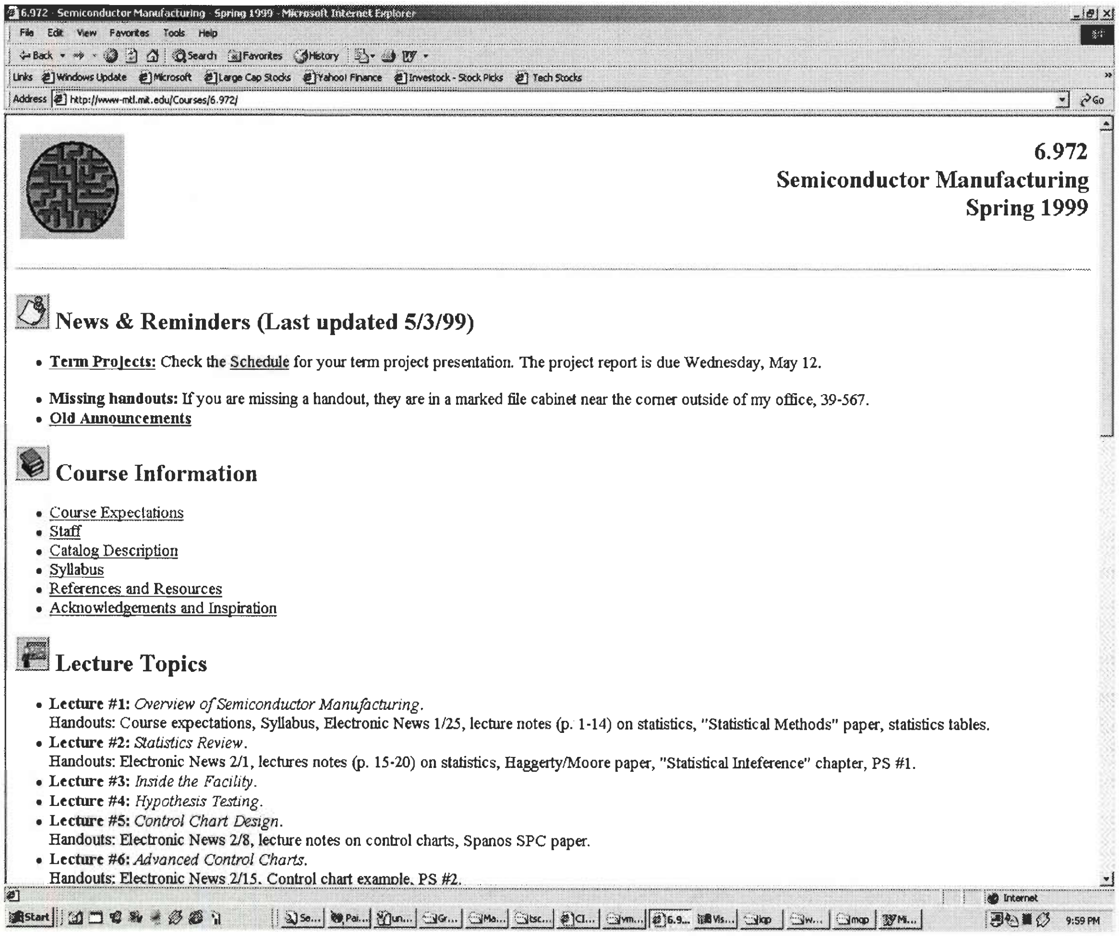
- Boston University
- Cornell University

- Dartmouth College
- NYU (New York University)
- RPI (Rensselaer Polytechnic Institute)
- University of Massachusetts Amherst
- University of Massachusetts Darthmouth
- MIT (Massachusetts Institute of Technology)
- University of Connecticut
- University of New Hampshire
- WPI (Worcester Polytechnic Institute)

All of the schools on this list have adapted to online long distance learning, but some of the universities and colleges have more advanced interactive educational systems then other schools. MIT (Massachusetts Institute of Technology), one of the top 10 schools in the United States, has very simple and standard online web modules. The course instructors usually make these standard modules themselves. The standard systems include the following features:

- Course Information or Syllabus
- Handouts
- Announcements
- Homework problems
- Course calendar
- Lab hours

Figure 1 Shows one of MIT's interactive web based modules.


6.972 - Semiconductor Manufacturing - Spring 1999 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Windows Update Microsoft Large Cap Stocks Yahoo! Finance Investock - Stock Picks Tech Stocks
Address http://www-mtl.mit.edu/Courses/6.972/
Go
Internet
Start
9:59 PM

**6.972**
**Semiconductor Manufacturing**
**Spring 1999**

**News & Reminders (Last updated 5/3/99)**

- **Term Projects:** Check the Schedule for your term project presentation. The project report is due Wednesday, May 12.
- **Missing handouts:** If you are missing a handout, they are in a marked file cabinet near the corner outside of my office, 39-567.
- **Old Announcements**

**Course Information**

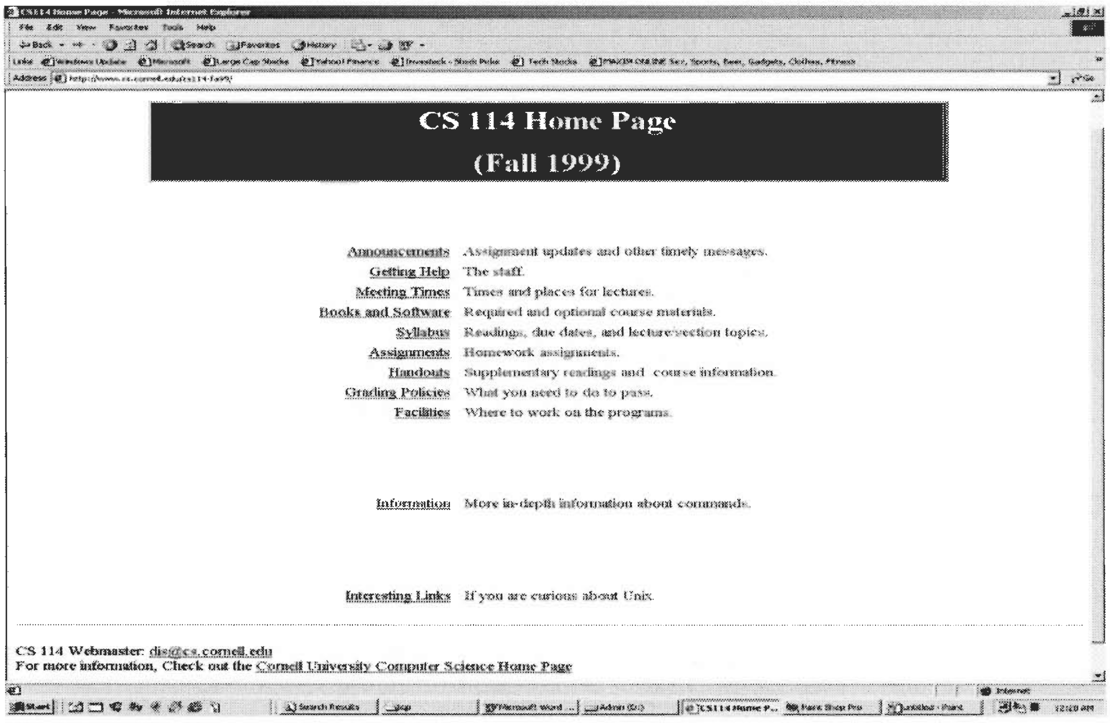
- Course Expectations
- Staff
- Catalog Description
- Syllabus
- References and Resources
- Acknowledgements and Inspiration

**Lecture Topics**

- **Lecture #1:** *Overview of Semiconductor Manufacturing.*
  Handouts: Course expectations, Syllabus, Electronic News 1/25, lecture notes (p. 1-14) on statistics, "Statistical Methods" paper, statistics tables.
- **Lecture #2:** *Statistics Review.*
  Handouts: Electronic News 2/1, lectures notes (p. 15-20) on statistics, Haggerty/Moore paper, "Statistical Inteference" chapter, PS #1.
- **Lecture #3:** *Inside the Facility.*
- **Lecture #4:** *Hypothesis Testing.*
- **Lecture #5:** *Control Chart Design.*
  Handouts: Electronic News 2/8, lecture notes on control charts, Spanos SPC paper.
- **Lecture #6:** *Advanced Control Charts.*
  Handouts: Electronic News 2/15. Control chart example. PS #2.

*Fig.1 - MIT electrical engineering course web wage*

Other schools that have the same simple but easy to use interactive pages include: Cornell University, Dartmouth College, NYU (New York University), and the University of New Hampshire. Some of the Colleges or Universities have added other modules but they are still very simple test links. Shown below are some of the course pages of the previously mentioned institutions.



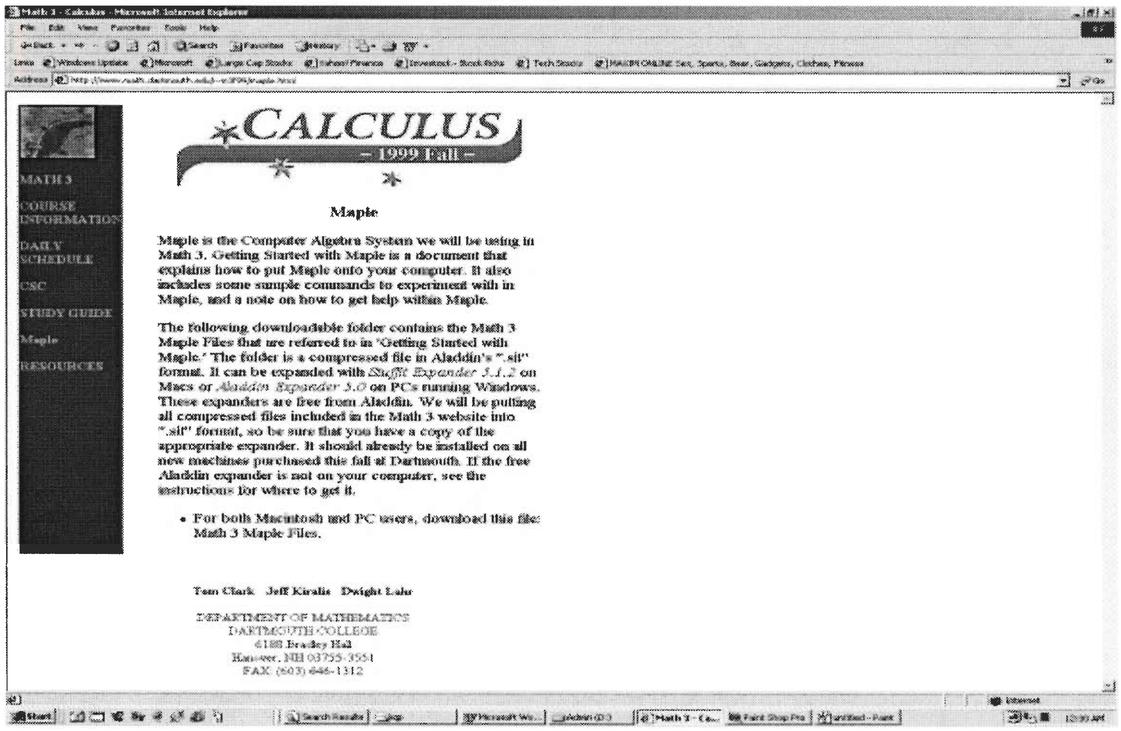
CS 114 Home Page

(Fall 1999)

| | |
|---|---|
| Announcements | Assignment updates and other timely messages. |
| Getting Help | The staff. |
| Meeting Times | Times and places for lectures. |
| Books and Software | Required and optional course materials. |
| Syllabus | Readings, due dates, and lecture/section topics. |
| Assignments | Homework assignments. |
| Handouts | Supplementary readings and course information. |
| Grading Policies | What you need to do to pass. |
| Facilities | Where to work on the programs. |
| Information | More in-depth information about commands. |
| Interesting Links | If you are curious about Unix. |

CS 114 Webmaster: dis@cs.cornell.edu
For more information, Check out the Cornell University Computer Science Home Page

*Fig. 2 - Cornell University; computer science course page*



**Maple**

Maple is the Computer Algebra System we will be using in Math 3. Getting Started with Maple is a document that explains how to put Maple onto your computer. It also includes some sample commands to experiment with in Maple, and a note on how to get help within Maple.

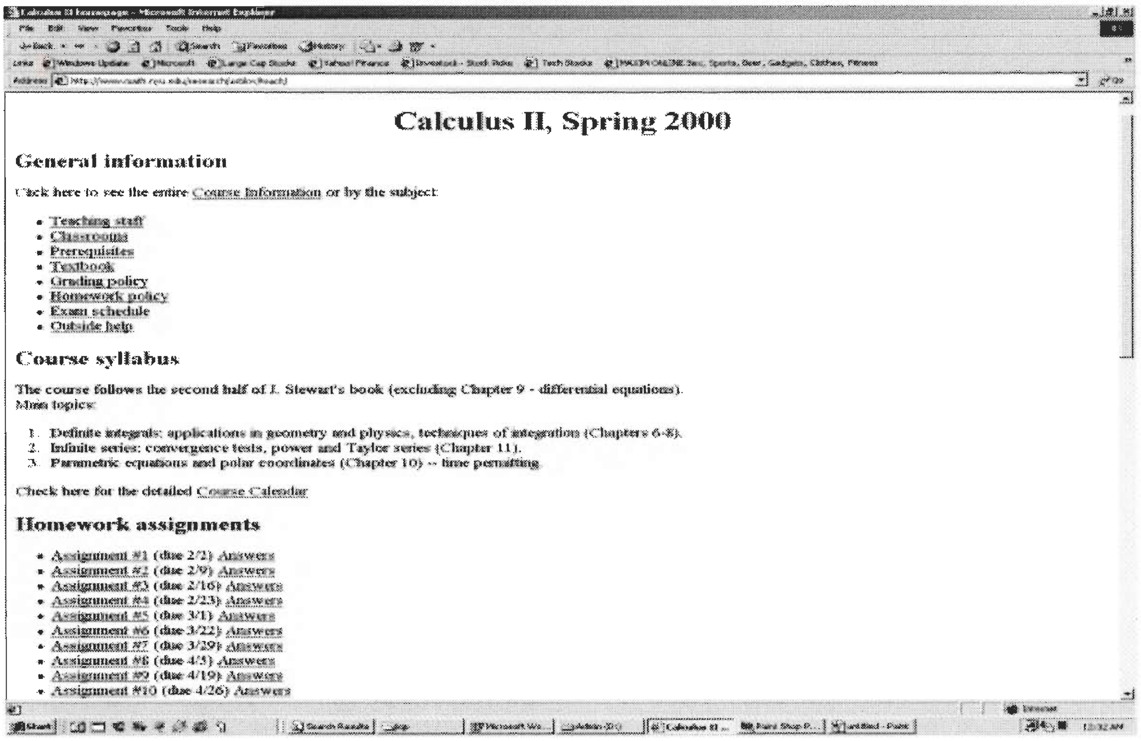
The following downloadable folder contains the Math 3 Maple Files that are referred to in 'Getting Started with Maple.' The folder is a compressed file in Aladdin's ".sit" format. It can be expanded with *Stuffit Expander 5.1.2* on Macs or *Aladdin Expander 5.0* on PCs running Windows. These expanders are free from Aladdin. We will be putting all compressed files included in the Math 3 website into ".sit" format, so be sure that you have a copy of the appropriate expander. It should already be installed on all new machines purchased this fall at Dartmouth. If the free Aladdin expander is not on your computer, see the instructions for where to get it.

- For both Macintosh and PC users, download this file: Math 3 Maple Files.

Tom Clark Jeff Kiralis Dwight Lahr

DEPARTMENT OF MATHEMATICS
DARTMOUTH COLLEGE
6188 Bradley Hall
Hanover, NH 03755-3551
FAX (603) 646-1312

*Fig.3 - Dartmouth College; Math course on the web*



Calculus II, Spring 2000

General information

Click here to see the entire Course Information or by the subject:

- Teaching staff
- Classrooms
- Prerequisites
- Textbook
- Grading policy
- Homework policy
- Exam schedule
- Outside help

Course syllabus

The course follows the second half of J. Stewart's book (excluding Chapter 9 - differential equations).
Main topics:

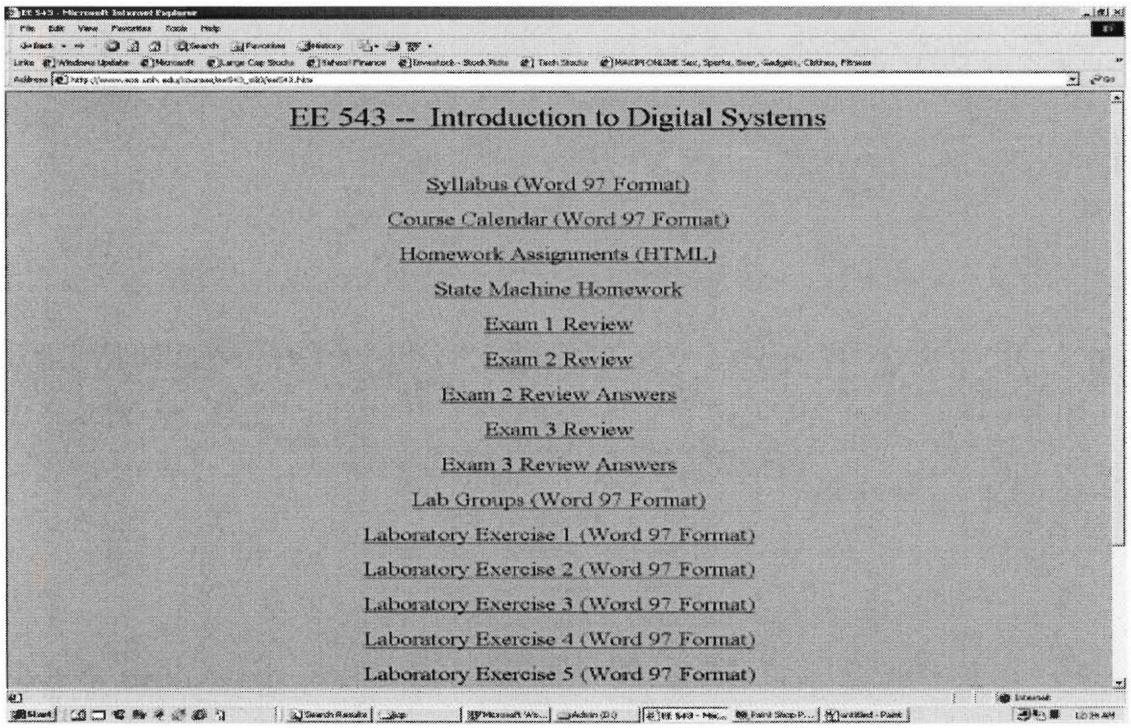
1. Definite integrals: applications in geometry and physics, techniques of integration (Chapters 6-8).
2. Infinite series: convergence tests, power and Taylor series (Chapter 11).
3. Parametric equations and polar coordinates (Chapter 10) -- time permitting

Check here for the detailed Course Calendar

Homework assignments

- Assignment #1 (due 2/2) Answers
- Assignment #2 (due 2/9) Answers
- Assignment #3 (due 2/16) Answers
- Assignment #4 (due 2/23) Answers
- Assignment #5 (due 3/1) Answers
- Assignment #6 (due 3/22) Answers
- Assignment #7 (due 3/29) Answers
- Assignment #8 (due 4/5) Answers
- Assignment #9 (due 4/19) Answers
- Assignment #10 (due 4/26) Answers

*Fig.4 - NYU (New York University); Calculus 2 web page*



EE 543 -- Introduction to Digital Systems

Syllabus (Word 97 Format)

Course Calendar (Word 97 Format)

Homework Assignments (HTML)

State Machine Homework

Exam 1 Review

Exam 2 Review

Exam 2 Review Answers

Exam 3 Review

Exam 3 Review Answers

Lab Groups (Word 97 Format)

Laboratory Exercise 1 (Word 97 Format)

Laboratory Exercise 2 (Word 97 Format)

Laboratory Exercise 3 (Word 97 Format)

Laboratory Exercise 4 (Word 97 Format)

Laboratory Exercise 5 (Word 97 Format)

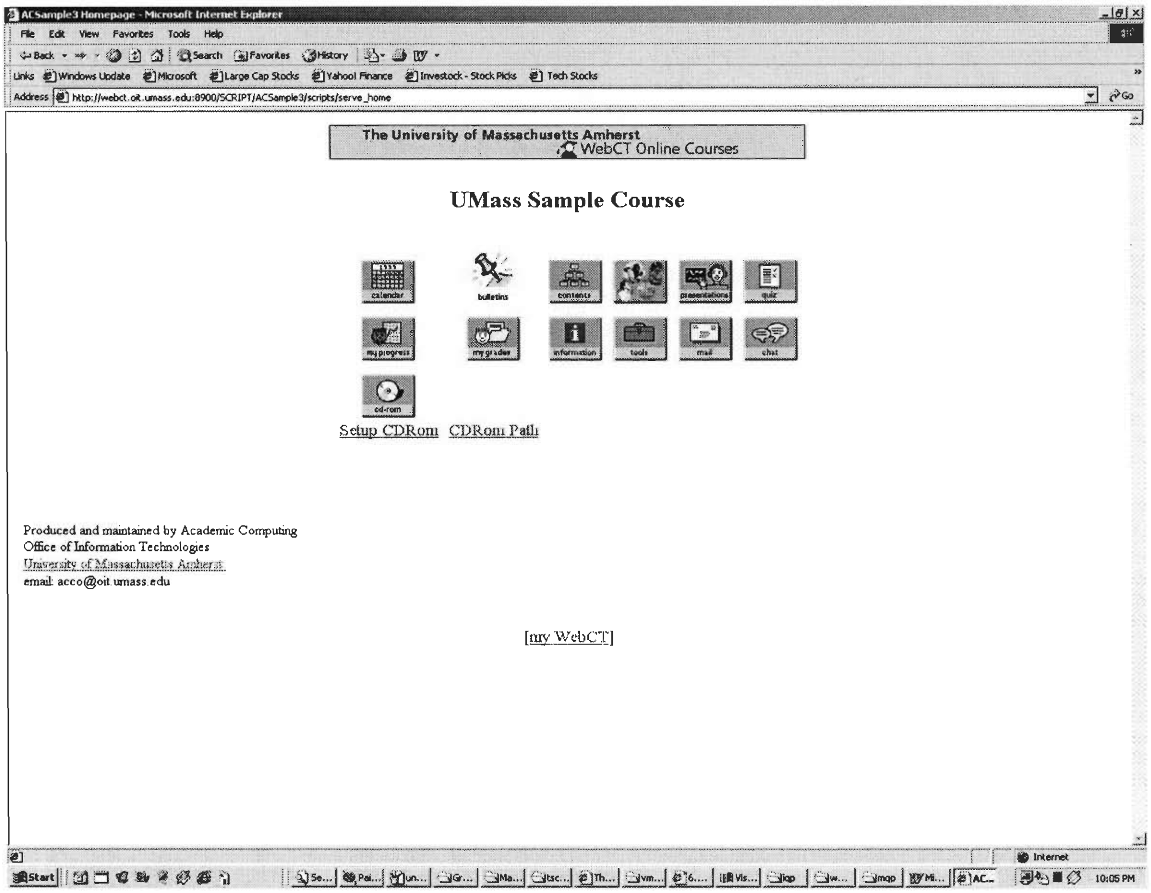
*Fig.5 - University of New Hampshire; Electrical Engineering Web Page*

Other schools have a more advanced systems which offer security systems to allow personalized settings and a login. These schools include Boston university, RPI (Rensselaer Polytechnic Institute), the University of Connecticut, and the University of Massachusetts Amherst have all adapted to the WebCT system. The WebCT system is a package designed to assist instructors in creating course web page sites. There are modules built in the system that instructor can use for any course that is going to be taught. These built in modules are:

- A Calendar – Instructor can post assignments and exams date. Student can also use the calendar as a scheduler.
- Bulletins – All users can post message and announcements.
- Content – similar to a syllabus
- Presentation - All users can present their projects on line using text or power point slides.
- Quiz – A quiz system where student can take an online quiz and then submit it to the instructor. The quiz is graded and can be posted the on the same page.
- My progress – Student profile and the amount of usage of WebCT
- My grades – post student homework, quiz, project, and exams grades
- Information – News and frequently asked questions about the course
- Tools - Simple tools that you can use on the site. These tools include a search engine to search the site contents, links so that a user can link outside sites to the WebCT page, a "compile" utility that allows users to print out sections of the course, and "Homepage" which allows a student to create their own personal web page.

- Mail – A private mail system which provides three main functions: the ability to send, read, and search for mail messages
- Chat – There are five different rooms a user can use for educational purposes
- CD-ROM – Allows users to access specific course files from a CD-ROM on the course server instead of downloading them from the Internet.

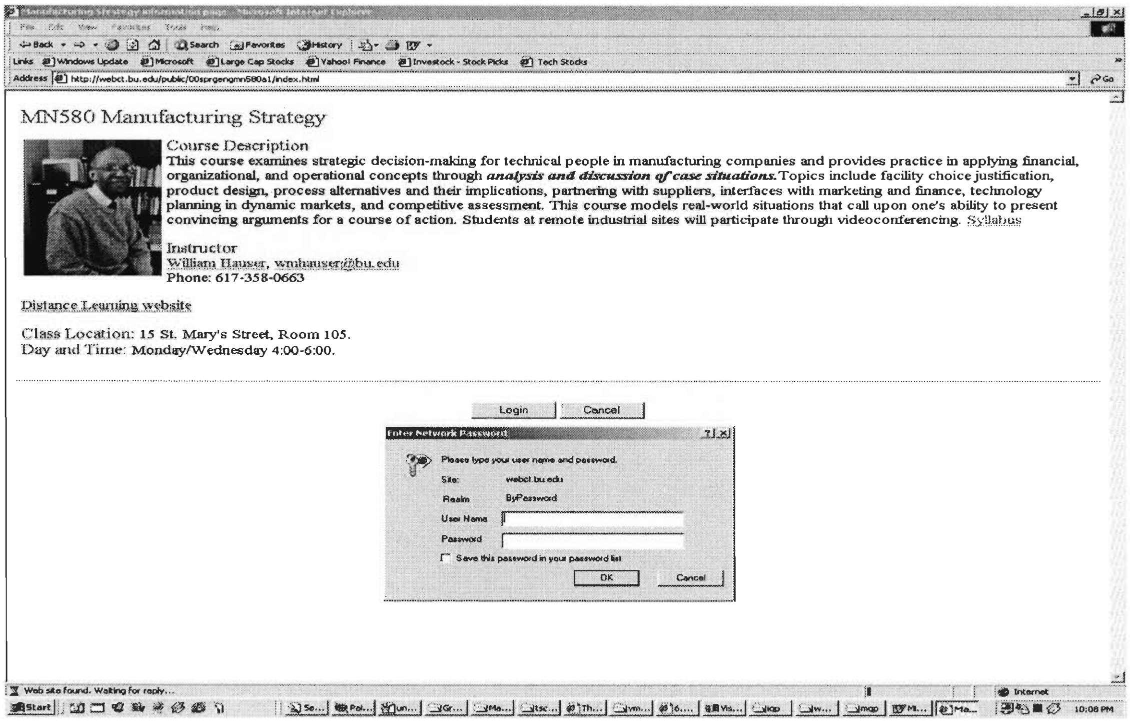
Examples of the WebCt system of different schools are shown below.


ACSample3 Homepage - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Windows Update Microsoft Large Cap Stocks Yahoo! Finance Investock - Stock Picks Tech Stocks
Address http://webct.oit.umass.edu:8900/SCRIPT/ACSample3/scripts/serve_home
Go
The University of Massachusetts Amherst
WebCT Online Courses
calendar
bulletins
contents
presentations
quiz
my progress
my grades
information
tools
mail
chat
cd-rom
Setup CDRom
CDRom Path
[my WebCT]
Internet
Start
10:05 PM

UMass Sample Course

Produced and maintained by Academic Computing
Office of Information Technologies
University of Massachusetts Amherst
email: acco@oit.umass.edu

*Fig.6 - Umass Amherst; WebCt system*



http://webct.bu.edu/public/00sprgengmn580a1/index.html

MN580 Manufacturing Strategy

Course Description
This course examines strategic decision-making for technical people in manufacturing companies and provides practice in applying financial, organizational, and operational concepts through ***analysis and discussion of case situations.*** Topics include facility choice justification, product design, process alternatives and their implications, partnering with suppliers, interfaces with marketing and finance, technology planning in dynamic markets, and competitive assessment. This course models real-world situations that call upon one's ability to present convincing arguments for a course of action. Students at remote industrial sites will participate through videoconferencing. Syllabus

Instructor
William Hauser, wmhauser@bu.edu
Phone: 617-358-0663

Distance Learning website

Class Location: 15 St. Mary's Street, Room 105.
Day and Time: Monday/Wednesday 4:00-6:00.

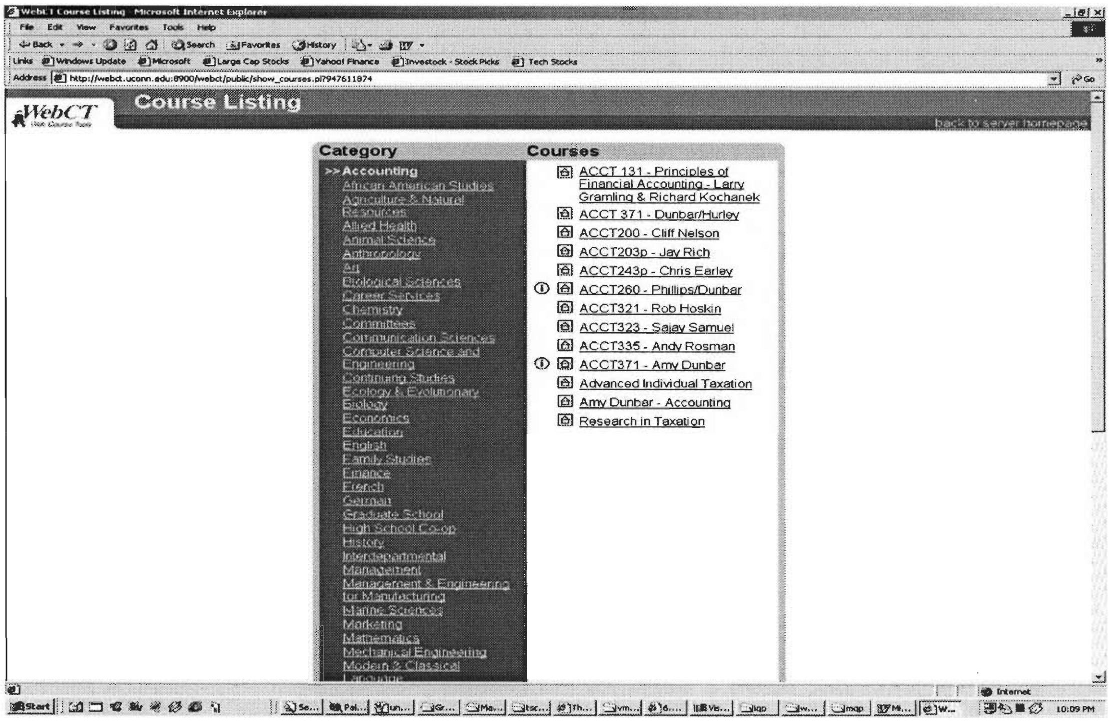
Enter Network Password

Please type your user name and password.

| | |
|---|---|
| Site: | webct.bu.edu |
| Realm | ByPassword |
| User Name | |
| Password | |

Save this password in your password list

*Fig.7 - Boston University; WebCT Login Screen*

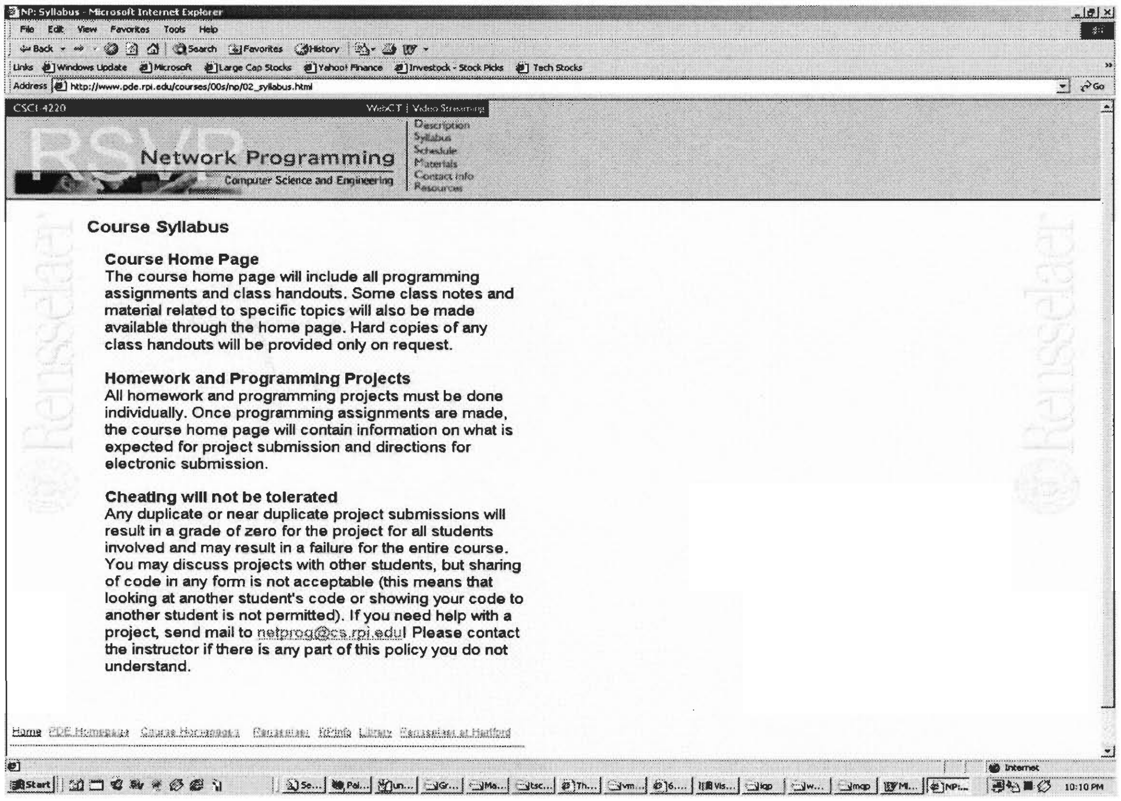


http://webct.uconn.edu:8900/webct/public/show_courses.pl?947611874

Course Listing

| Category | Courses |
|---|---|
| >> Accounting | ACCT 131 - Principles of Financial Accounting - Larry Gramling & Richard Kochanek |
| African American Studies | ACCT 371 - Dunbar/Hurley |
| Agriculture & Natural Resources | ACCT200 - Cliff Nelson |
| Allied Health | ACCT203p - Jay Rich |
| Animal Science | ACCT243p - Chris Earley |
| Anthropology | ACCT260 - Phillips/Dunbar |
| Art | ACCT321 - Rob Hoskin |
| Biological Sciences | ACCT323 - Sajay Samuel |
| Career Services | ACCT335 - Andy Rosman |
| Chemistry | ACCT371 - Amy Dunbar |
| Committees | Advanced Individual Taxation |
| Communication Sciences | Amy Dunbar - Accounting |
| Computer Science and Engineering | Research in Taxation |
| Continuing Studies | |
| Ecology & Evolutionary Biology | |
| Economics | |
| Education | |
| English | |
| Family Studies | |
| Finance | |
| French | |
| German | |
| Graduate School | |
| High School Co-op | |
| History | |
| Interdepartmental | |
| Management | |
| Management & Engineering for Manufacturing | |
| Marine Sciences | |
| Marketing | |
| Mathematics | |
| Mechanical Engineering | |
| Modern & Classical Language | |

*Fig.8 - University of Connecticut; Courses on WebCt*


NP: Syllabus - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Windows Update Microsoft Large Cap Stocks Yahoo! Finance Investock - Stock Picks Tech Stocks
Address http://www.pde.rpi.edu/courses/00s/np/02_syllabus.html
CSCI 4220
WebCT | Video Streaming
Network Programming
Computer Science and Engineering
Description
Syllabus
Schedule
Materials
Contact info
Resources
Home PDE Homepage Course Homepages Rensselaer RPInfo Library Rensselaer at Hartford
Start
10:10 PM

Course Syllabus

Course Home Page
The course home page will include all programming assignments and class handouts. Some class notes and material related to specific topics will also be made available through the home page. Hard copies of any class handouts will be provided only on request.

Homework and Programming Projects
All homework and programming projects must be done individually. Once programming assignments are made, the course home page will contain information on what is expected for project submission and directions for electronic submission.

Cheating will not be tolerated
Any duplicate or near duplicate project submissions will result in a grade of zero for the project for all students involved and may result in a failure for the entire course. You may discuss projects with other students, but sharing of code in any form is not acceptable (this means that looking at another student's code or showing your code to another student is not permitted). If you need help with a project, send mail to netprog@cs.rpi.edu! Please contact the instructor if there is any part of this policy you do not understand.

*Fig.9 - RPI (Rensselaer Polytechnic Institute); Course syllabus in WebCt*

The aforementioned schools have different ideas as to the extent and depth of web based materials. The interactive web based modules found at these different schools vary from simple informational systems in which users can retrieve course data, to fully interactive web sites in which users may communicate with one another in a real-time format.

The decision to go with one system or another is based often upon the needs of the course. If, for instance, a course includes a large amount of group work, then a chat room could be of tremendous benefit to students in the course who need to communicate with their project partners. In other circumstances however, the extra features offered by an interactive system are often not necessary and not worth the effort to create.

## *3.2.1 Evaluation and Comparison of the Schools*

The following chart attempts to rank the following institutions base upon a series of criteria. These criteria includes security, Accessibility, speed, graphical user interface, and features. A 1 to 10 scale is used in attempt to grade each of these features where a 1 indicates a bad score and a 10 indicates an excellent score. The total score gives some kind of indication of the quality of the interactive web modules the school offers.

*Table.1 - Evaluation of features offered by schools*

| | Security | Easy to Access | Connection Speed | Graphical user interface | Features | Total score |
|---|---|---|---|---|---|---|
| Boston University | 5 | 4 | 6 | 7 | 7 | 29 |
| Cornell University | 0 | 8 | 7 | 5 | 3 | 23 |
| Dartmouth College | 0 | 6 | 6 | 7 | 4 | 23 |
| New York University | 0 | 7 | 7 | 6 | 3 | 23 |
| Rensselaer Polytechnic Institute | 7 | 8 | 8 | 9 | 8 | 40 |
| Umass Dartmouth | 0 | 5 | 8 | 5 | 5 | 23 |
| Umass Amherst | 9 | 7 | 7 | 10 | 10 | 43 |
| MIT | 0 | 7 | 9 | 8 | 5 | 29 |
| University of Connecticut | 6 | 6 | 8 | 7 | 5 | 32 |
| University of New Hampshire | 0 | 9 | 7 | 4 | 4 | 24 |

The next table gives an indication of what features each school offers.

*Table.2 - Features offered by different schools.*

| | Announcement | Assignments | Bulletin board | File upload | chat | Quiz system | grades | Secure system |
|---|---|---|---|---|---|---|---|---|
| Boston University | X | X | X | X | X | | X | X |
| Cornell University | X | X | | | | | | |
| Dartmouth College | X | X | | | | | | |
| New York University | X | X | | | | | | |
| Rensselaer Polytechnic Institute | X | X | X | X | X | | X | X |
| Umass Dartmouth | X | X | | | | | | |
| Umass Amherst | X | X | X | X | X | X | X | X |
| MIT | X | X | | | | | | |
| University of Connecticut | X | X | X | X | X | X | X | X |
| University of New Hampshire | X | X | | | | | | |

### *3.2.2 Systems and Their Features*

The following table compares the major features offered by different interactive web module software.

**Students**

*Table.3 - Features offered for students by different systems*

| | Black Board | ReCourse | ILN | WebCT | TopClass | CyberProf |
|---|---|---|---|---|---|---|
| Announcements | X | | X | | X | |
| Assignments | X | | X | | X | X |
| Bulletin Board/Newsgroup | X | X | X | X | X | X |
| Chat | X | X | X | X | | |
| Grades | X | X | X | X | | X |
| Quiz System | X | X | X | X | X | X |
| ReTargetable | X | X | | | | |
| Site Map | X | X | | | | |
| Site Search | X | X | | X | | |

**Instructors**

*Table.4 – Features offered for professors by different systems*

| | Black Board | ReCourse | ILN | WebCT | TopClass | CyberProf |
|---|---|---|---|---|---|---|
| Announcements Posting | X | | X | | X | |
| Assignments | X | X | X | X | X | X |
| Bulletin Board | X | X | X | X | X | X |
| Chat | X | X | X | X | | |
| Course Backup | | | | X | | |
| Features Admin* | | X | | X | | |
| File Upload | X | X | | | | |
| Grades | X | X | X | X | X | X |
| Quiz System | X | X | X | X | X | X |
| Site Map | X | X | | | | |
| User Administration | X | X | X | X | X | |

## *3.2.3 System and Feature Analysis*

The educational systems mentioned above all have the same basic elements. The common elements included in these modules and other educational web-sites include the following:

- Announcements
- Bulletin Board
- Syllabus
- Homework Assignments
- Sample Exams (non-interactive)
- Online Chat
- Secure Grading System
- Group Email
- Glossary of terms

## *3.2.4 Elements in a Web Based Module*

The following is a description of some of the more important features that I am implementing.

Bulletin Board

The bulletin board is a means of communication between the instructor and his or her students. Students may post questions and comments to the instructor or to the entire class. The instructor may post announcements on the bulletin board as well. The instructor should also have the option of emailing the entire class, a specified group of students, or an individual student.

## Chat-room

The chat room will provide real time interaction between the instructor and the students, as well as between the students themselves. The instructor may hold office hours in the chat room for students who may not be able to attend the regular office hours. Instructors and students will have the same interface to the chat room. Users will connect to the server and then the chat room screen will appear. The chat room screen will include a list of connected users, the chat room log, and a text box for the user to enter a message. The user will have the option of broadcasting a message to everyone in the chat room or to a specific user in the chat room.

## Quiz System

The quiz system may be useful for a web-based module in several ways. An online quiz system may be similar to take home exams. In addition, the quiz system may act as a practice exam for the students or as a self-evaluation throughout the course.

Templates will allow the instructors to create quizzes and examinations online. The instructor may create quizzes through a few simple steps. First, the instructor selects whether the questions will be multiple choice, true/false, fill-in-the-blank, matching, or short answer questions. After selecting the type of question, an appropriate template will be generated depending on what type of question it is. The instructor will have the options of creating sections for a particular set of questions, deleting a single question or deleting questions in a section.

## Database Generation

Database generation makes tables with information relevant to the class. This function is particularly useful when examining test results from the whole class so the students can examine and study the results of their classmates.

The instructor can create and modify the data in a table. When a cell in the table is clicked and highlighted, a pop up message box will appear for the instructor or student to modify the contents of the cell.

## Experimental Simulations

Experimental Simulations serve to reinforce concepts from the classroom. Templates for this element would be virtually impossible to implement because of the endless simulation possibilities. However, with some guidance, a specific simulation could be designed for web-based module.

Graphing is one of the functions that could be implemented for the experimental simulation. For example, a possible model to simulate would be a spring. The student may modify parameters of the spring and see how each parameter affects the elasticity of the spring. A graph could be used to show the increasing or decreasing elasticity.

## Interactive Glossary

An interactive glossary is useful for high level courses where the vocabulary is very complicated. All the words in the glossary are stored in a database that is modified by the instructor only. The instructor can add or delete words from the glossary at anytime. The students can search for a particular word in the glossary or browse the index for all the available words.

Surveys

Surveys are often conducted by instructors to obtain a consensus or opinion from the class. For certain types of surveys, the instructor would be allowed to compute the mean, median, and standard deviation with the click of a button. This feature would save time for the instructor and his assistants. For surveys where the student may be asked to choose a particular project or assignment, the instructor can view the results online and determine the project assignment for each student or student group and then email the student or group regarding his decision.

Graphing Utility

A graphing utility may be implemented to help students visualize certain equations used in a course. This utility will allow students to "plug-in" equations from class and see the graphical representation of the data. Students will be able to graph all two-variable functions.

## 3.3 Internet Learning at WPI

Using the Internet to convey an idea is very common is today's society. Being a technical school, WPI has also followed the trend of using the Internet for educational purposes. Many professors at WPI use the Internet to communicate and interact with their students. Information on a particular course is posted on the web so students in the class and perhaps potential students can look at the information at any time. In addition, much paper is saved with the use of the Internet because there will be no unnecessary handouts.

The general format of a course web page includes the syllabus, homework assignments, exams, help sessions and announcements. Information often included in a

WPI course web page includes the syllabus, schedule, textbook, homework assignments, course description, help session times, grading policy, class notes, announcements, exams, and miscellaneous handouts.

## 3.4 Summary

By evaluating existing course web pages, some distinct trends in design and organization can be readily recognized. The web page is often divided into sections, which can be linked from the main page. Oftentimes, each linked section is presented as a separate html page. However, in simpler web pages, they are presented as sections from the main html page. In more elaborate cases, the web page uses frames to list the sections. Most of the web pages use minimal graphics. Bullets are effectively used to list information. Section heads are often listed using bullets. Tables are often used to present information regarding class schedules, and help session times. In some cases, all the documents on the web page were *pdf* files(footnote – what is a pdf file), which meant that the viewer must have a *pdf* viewer in order to see the documents. Some course web pages also include secure pages that can only be accessed by the students in the class. In nearly all the web pages reviewed, information was presented to the students and viewers clearly. The web pages generally provided sufficient resources for students, such that if the student missed a class, he or she does not have to be alarmed. Homework assignments and class handouts were often posted on the web page so students can access them. In addition, last minute announcements can also be made via the Internet. Graphics for most of the web pages reviewed were kept to a minimal probably in an attempt to decrease the download time for the viewer. Furthermore, since the web pages must be changed or updated each term, spending excessive time on web page graphics would not be practical for the instructor and his or her assistants.

# 4.0 Development of Communication Modules

After researching what other interactive web-based educational systems have to offer the requirements for a web based interactive module set at WPI was analyzed. To determine what would be required of WPI's web based modules, other software was examined and used to create a new set of web based interactive modules. Some of the software had a great interface, making for simple user interaction. The down fall of this was that it was difficult to implement new features or improve on existing ones. There were simple implementations of the same software but they lacked the nice user interface. As a result, some of the simpler versions of these web-base modules have been improved upon. This allows for an easily coded interface and to a functional user interface. To improve the user interface, make future modification easier, and to be able to improve the interactive web-based educational systems, most of the web-base modules were implemented using ASP (Active Server Pages).

The intended users of these types of interfaces are students currently enrolled in and attending a class in which the web based interactive modules are used as a teaching aid and the professors or instructors of the course. The use of these web modules is not complex. Oftentimes help files are offered to aid the user in understanding the software. However, since the interactive modules are an internet based technology, a good understanding of the Internet is vital to proper usage and understanding of the web modules. At WPI, much of the student body is comprised of individuals with a technical background. This makes acceptance and usage of these interactive modules much easier at WPI.

## 4.1 Programming Languages

Before any actual implementation of this project could be started, it was necessary to investigate the different tools that were available to me. Since this was a web-based application, some of the more popular languages available to web developers today were looked at first. Each of the languages that were focused on had their own strengths and weaknesses. It was these characteristics that helped me to determine which of the languages were most suited to the different features of our application. The languages that were investigated were HTML, Java, Perl, ASP, and VRML. This section will describe each language in detail and its possible applications.

### *4.1.1 HTML*

Hyper Text Markup Language (HTML) is the defacto standard for publishing hypertext documents, more commonly known as "web pages", on the World Wide Web. It is based on a more complicated markup language called SGML or "Standardized Generalized Markup Language." The HTML is a special markup language you use to create Web documents with hyperlinks ( links to other Web documents). The HTML is nit a programming language, but a set of rules used to format a Web document. When you create a hypertext document using HTML, you must follow a set of rules. You have the option to embed many types of formatting information into your Web document. The formatting information you provide in a Web document describes the structure of the information in the document to the Web browsers, which in turn display the document within a window on your screen.

### *4.1.2 Java*

For years, people have had a hard time viewing image files on the Internet due to the slow transmission rates of most modem-base connections. Because of the limited bandwidth (due to slow transmission line), it was nearly impossible to animate a Web site. Java is a programming language, developed by Sun Microsystems, in which programmers can create applets (small applications) that the server downloads to the browser which, in turn, runs the applet on your computer. Using Java, programmers can create animation that move or spin object, multimedia applets that plays sounds and music, as well as an unlimited number of conventional applets that may display stock information, process customer orders, and more. Java programming language also provides security features to restrict user and programmer from getting full control of the executable files when downloading.

### *4.1.3 Perl*

Perl or by its full name, "Practical Extraction and Report Language" was originally created as a "glue" language for the UNIX operating system. This means Perl was used by systems administrators to get different programs and features of the operating system to work together easily. As a result of its beginnings, Perl was first popular among systems administrators and later web developers because of its ability to process text quickly and easily. This is something that other popular languages such as "C" lacked.

Later as the World Wide Web became more popular, a growing number of web developers learned of Perl's powerful text processing capabilities. It was this strength of Perl that quickly made it the language of choice for CGI or "Common Gateway Interface" scripting. A CGI script is a program written in any supported language, some examples include C and Perl, that is usually run on the server in response to a user action such as clicking the "Submit" button on an HTML form.

"Scripting" as it relates to the World Wide Web is used to create dynamic content, which generally takes the form of a web page, which is created on the fly and is affected by previous user input of some type. A typical Perl CGI script would process the incoming data from a form or another script, add the information to a file or database, and possibly email the user a message telling them that their information was correctly received. A skilled programmer can do all of that in about ten lines of Perl. This shows Perl's inherent power, which is a big advantage when attempting to create a complex application.

While being a powerful language has its advantages there is always a tradeoff with complexity. The more powerful a language is, the more complex it tends to be, thus making it more difficult to learn. Perl includes elements from both worlds allowing programmers to use the simple features of Perl without having to understand the more complex constructs to accomplish something useful. This makes Perl easy to learn, which is another advantage for my project.

The main disadvantage, of using Perl as one of the development languages in this project, was that the group was inexperienced in Perl programming. While this is a significant obstacle, it is far less an issue with a language such as Perl since one of its strengths is the ease of which it can be learned. By starting off with some simple programs and using the wealth of resources freely available on the web, such as example code and tutorials, learning the Perl language is a manageable proposition.

### *4.1.4 ASP*

"Active Server Pages" are another way to create dynamic content. They are similar to standard HTML pages except for one major difference that being the role of the web server. When dealing with a normal "static" HTML document there are three basic steps that take place. First, the web browser sends a request for a page to the web server. Second, the web server finds the requested page and finally sends the page back to the browser, which will then display it on the screen. In this case the browser will always display the page exactly the same way since the server it sending it the same information every time. This is where the difference between HTML and ASP becomes apparent.

When dealing with a "dynamic" Active Server Page there are four steps that must take place. First the web browser sends a request for the page to the web server. Second, the web server locates the page. Third, the web server now modifies the page according to any ASP instructions that are included in it. Finally, the now modified and thus "dynamic" page is sent back to the browser, which then displays it.

### *4.1.5 VRML*

"Virtual Reality Modeling Language" or VRML is a language for describing three-dimensional worlds and objects in those worlds. The VRML language is interpreted by a VRML viewer, which is often a web browser "plug-in" application. This setup allows web users to view the 3D content of a site from their standard web browser and does not require any specialized software other than the "plug-in." Such content will often include tours of a building or just particular objects of interest. In a VRML world the user is able to navigate through and sometimes interact with their virtual surroundings. An example of VRML behavior would be a VRML representation of a cube, which the user is able to rotate and move around. This allows a user to glean far more information about an object that by two-dimensional images. It also allows users to observe objects that may not be visible to them such as a DNA chain.

## 4.2 Interactive Web Based Modules

Development of test modules followed a distinct pattern. First, what would be required of the test modules and their appearance on the web site was decided upon. At this juncture, sections of the interactive web modules were coded using the different web programming languages.

### *4.2.1 Flow Diagram of the Interactive Web Based Modules System*

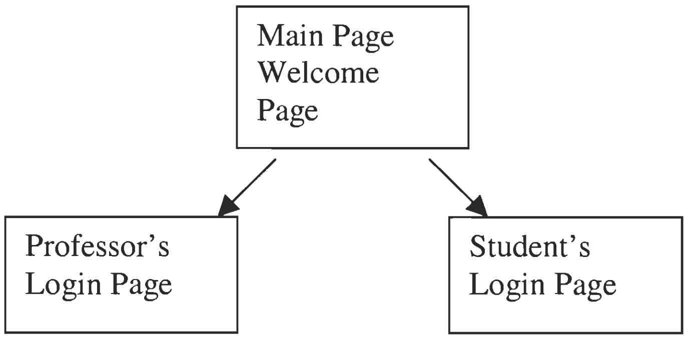


*Fig.10 - Block diagram of main page connecting to the different login page of the user choice.*

## *4.2.2 Professor's Login*

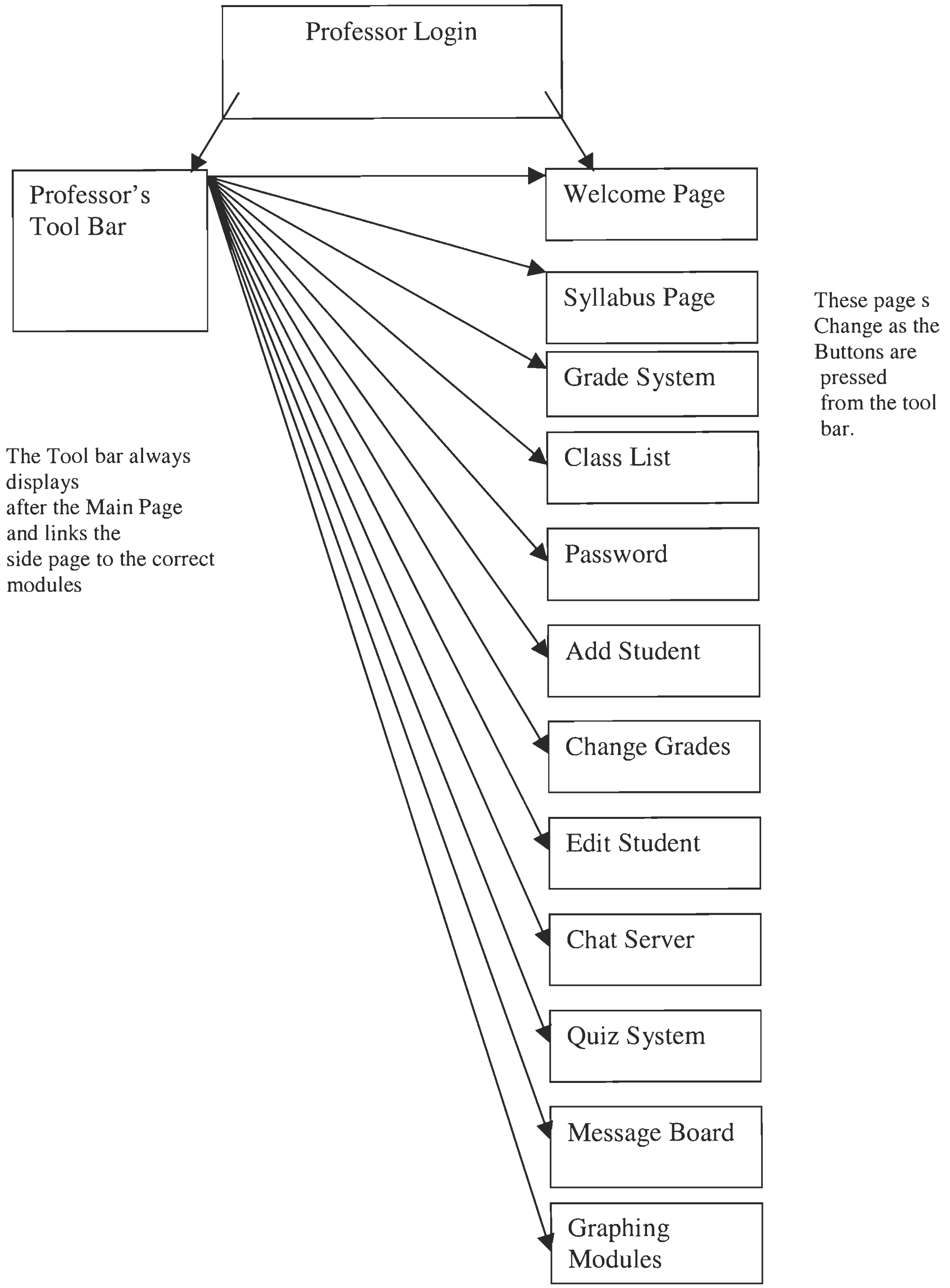
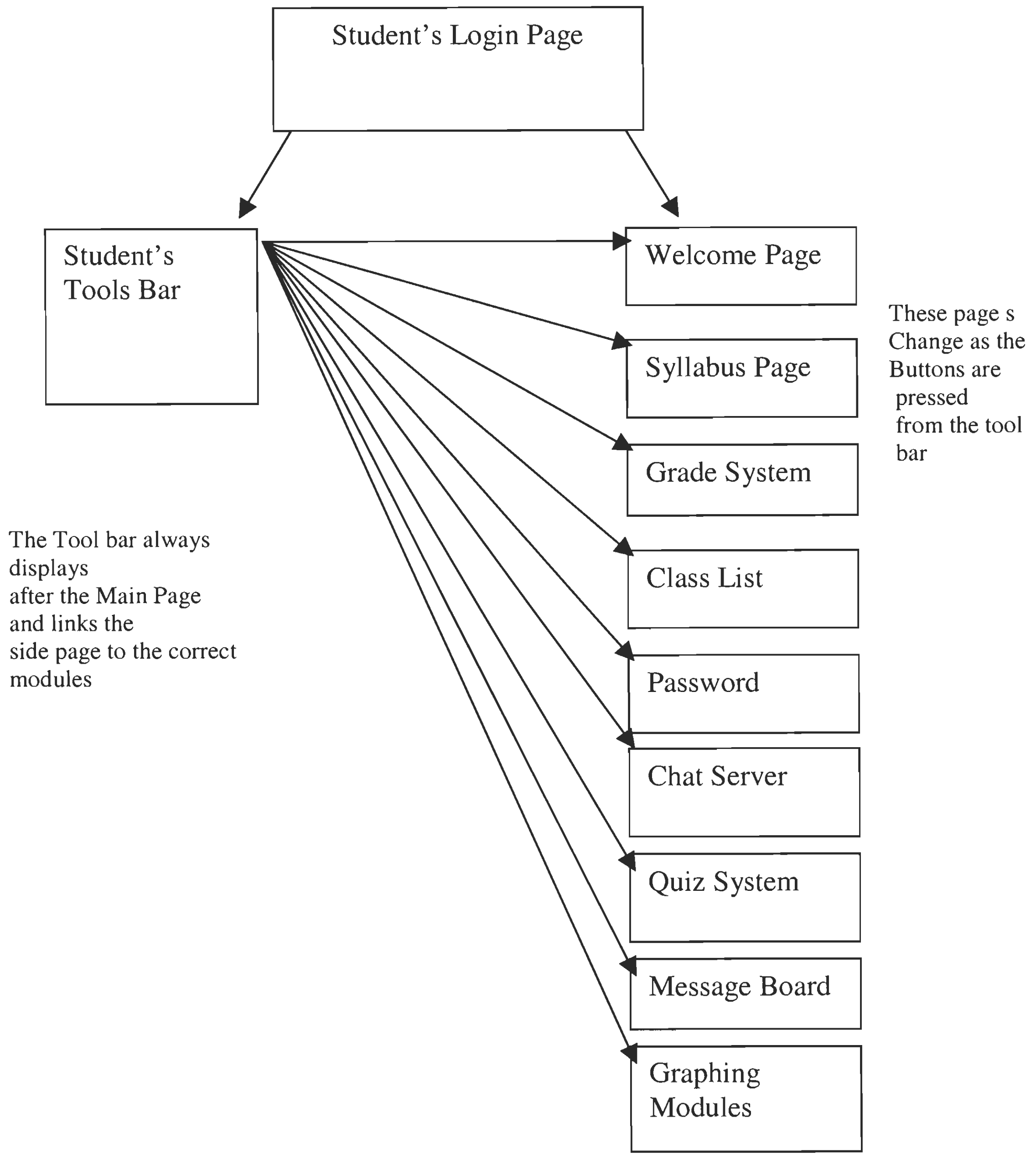


*Fig.11 – The Block diagram of the page will appear when the student login. The Selection will then be made on the student's tool bar.*

## *4.2.3 Student's Login*



Student's Login Page
Student's Tools Bar
Welcome Page
Syllabus Page
Grade System
Class List
Password
Chat Server
Quiz System
Message Board
Graphing Modules
These page s Change as the Buttons are pressed from the tool bar
The Tool bar always displays after the Main Page and links the side page to the correct modules

*Fig.12 - Block diagram of page will appear will the student is login and make the selection on the student tools bar.*

## *4.2.4 Web Module Files and Locations*

*Table.5 - Interactive web-based modules names and where it is located in the root directory.*

| **Page** | **Location** | **File name** |
|---|---|---|
| Main Page | .\default.html | default.html |
| Professor's login | .\main\instructorlogin.asp | instructorlogin.asp |
| Student login | .\main\studentlogin.asp | studentlogin.asp |
| Professor Tool Bar | .\main\professor\instructorbar.asp | instructorbar.asp |
| Student Tools Bar | .\main\student\studentbar.asp | studentbar.asp |
| Welcome Page | .\main\student\studentmain.asp | studentmain.asp |
| Syllabus Page | .\syllabus.html | syllabus.html |
| Grade Page | .\main\student\displaygrade.asp | displaygrade.asp |
| Class List | .\main\student\classinfo.asp | classinfo.asp |
| Password | .\main\professor\passwordchange.asp | passwordchange.asp |
| Add Student | .\main\professor\ addstudent.html | addstudent.html |
| Change Grades | .\main\professor\ changegrades.asp | changegrades.asp |
| Edit Student | .\main\professor\editstudent.asp | editstudent.asp |
| Chat Page | .\chatframes.html | chatframes.html |
| Quiz Page | .\quiz.html | quiz.html |
| Message board | .\messageboard.html | messageboard.html |
| Graphing modules | .\graph.html | graph.html |

## *4.2.5 Login Checker*

The Login checker was done with HTML and ASP. It works by using ASP code that takes a user entered login name and password, opens up a Microsoft Access data base (which contain all the registered user names and their passwords), and it compares the user name and password to those in the data base. If a match is found, the user may enter the web based interactive modules pages. In a case where an incorrect password or user name is entered, the modules raise an error message. A picture of a login screen is shown below.

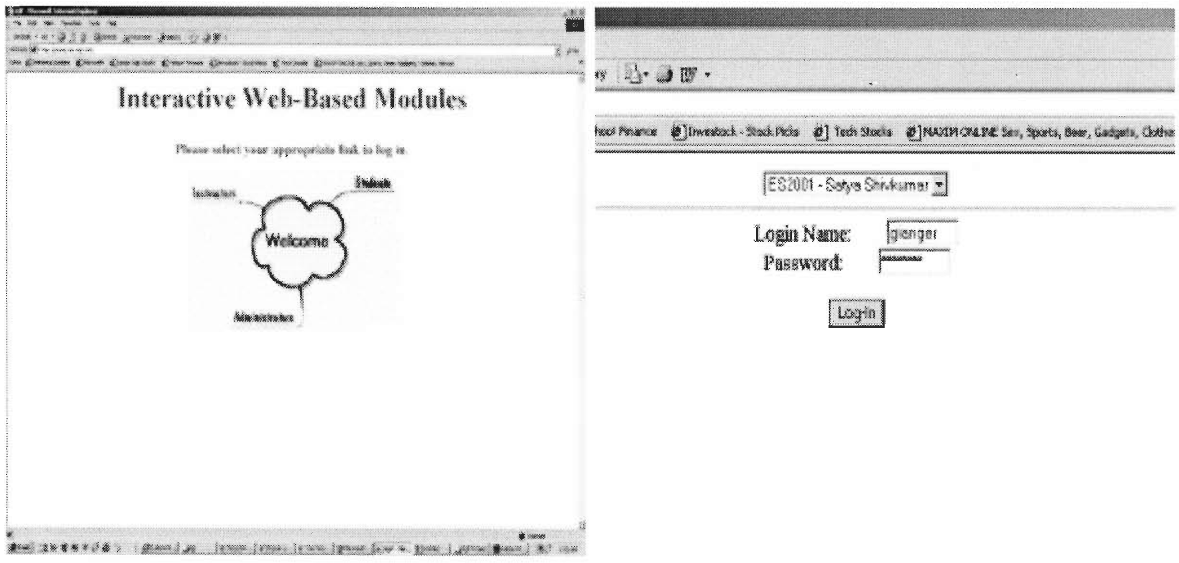


*Fig.13 - Picture of the main page and how instructor can login to the system for the interactive web-based modules page.*

Shown below is a flow chart of the login sequence followed to facilitate entry into the web module system.

Login checker

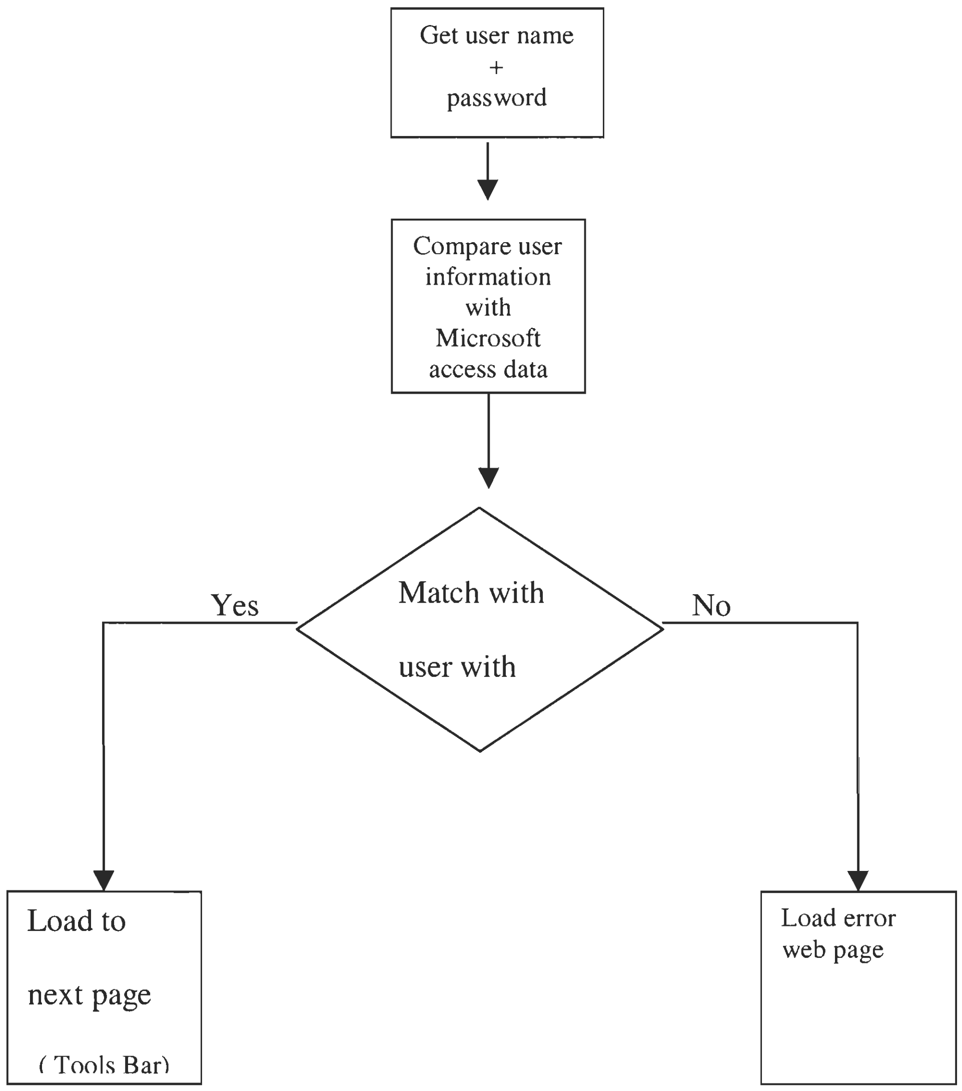
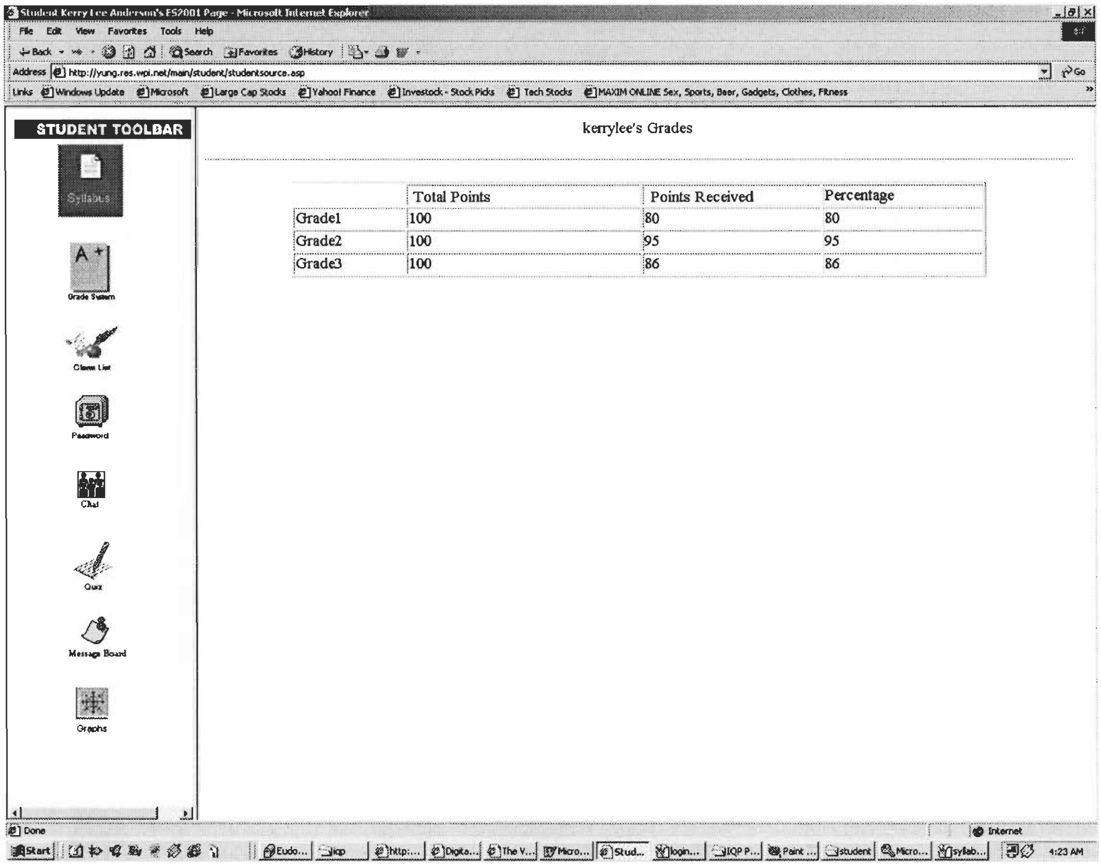

Get user name
+
password
Compare user information with Microsoft access data
Match with user with
Yes
No
Load to next page ( Tools Bar)
Load error web page

*Fig.14 - Flow chart for the login checker*

## 4.2.6 Secure Grading

The secure grading system was implemented with ASP, using the special ASP function to open a Microsoft Access data base to get the grade information for different users. For it to be secure, the user must login to check the grade. The grading page is also checked by another ASP function to ensure that the user is still active. If there is no active on the part of the user in a given time, the module times out and the page is no longer accessible. In this case, the user has to login again to check the grade. A picture of how the grades are being displayed is shown in the next figure.


Student Kerry Lee Anderson's ES2001 Page - Microsoft Internet Explorer
http://yung.res.wpi.net/main/student/studentsource.asp
STUDENT TOOLBAR
Syllabus
Grade System
Class List
Password
Chat
Quiz
Message Board
Graphs

kerrylee's Grades

| | Total Points | Points Received | Percentage |
|---|---|---|---|
| Grade1 | 100 | 80 | 80 |
| Grade2 | 100 | 95 | 95 |
| Grade3 | 100 | 86 | 86 |

*Fig.15 – Presentation of grades on the screen grading page.*

Shown here is a flow chart of the sequence used to retrieve grades in a secure fashion.

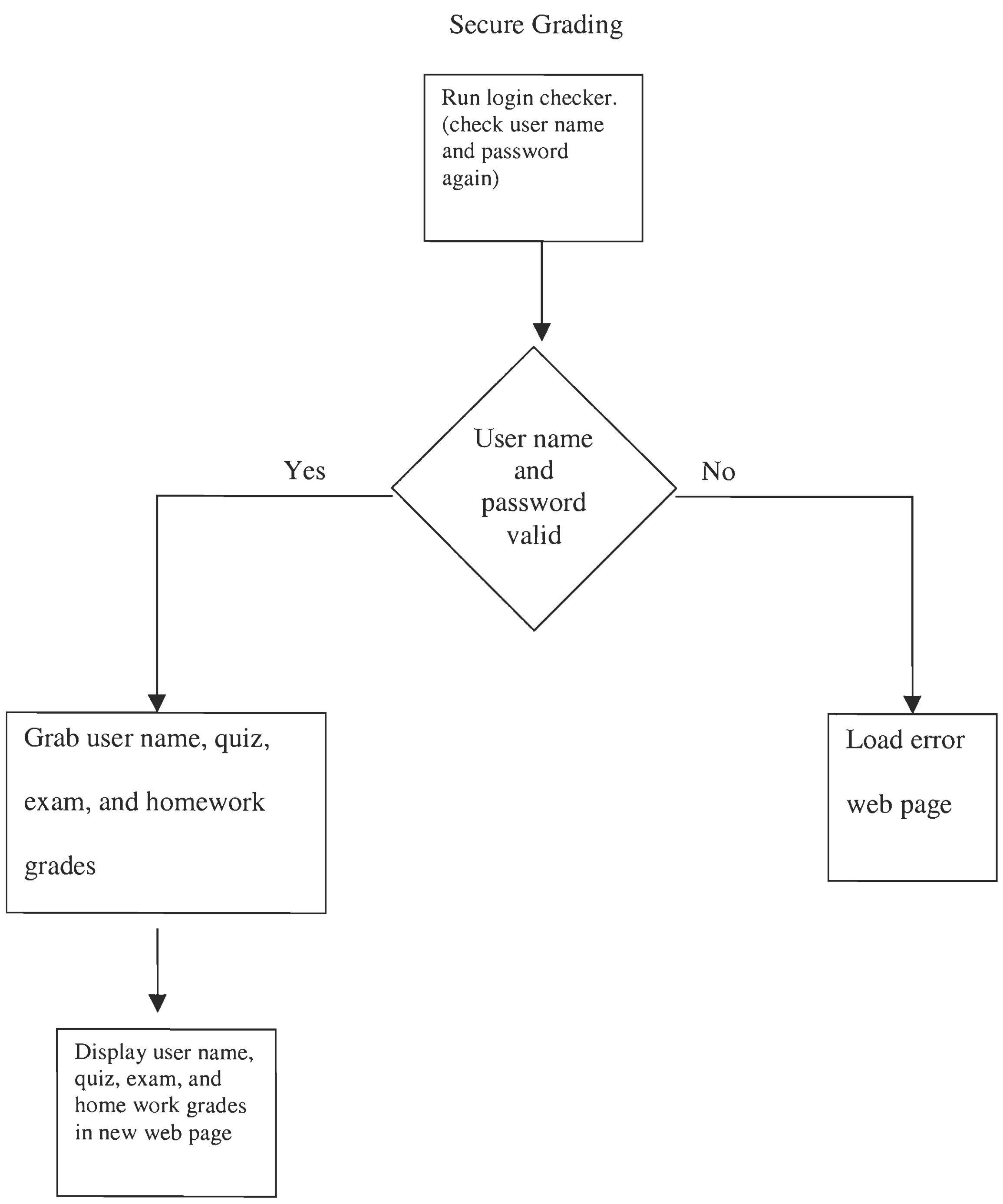
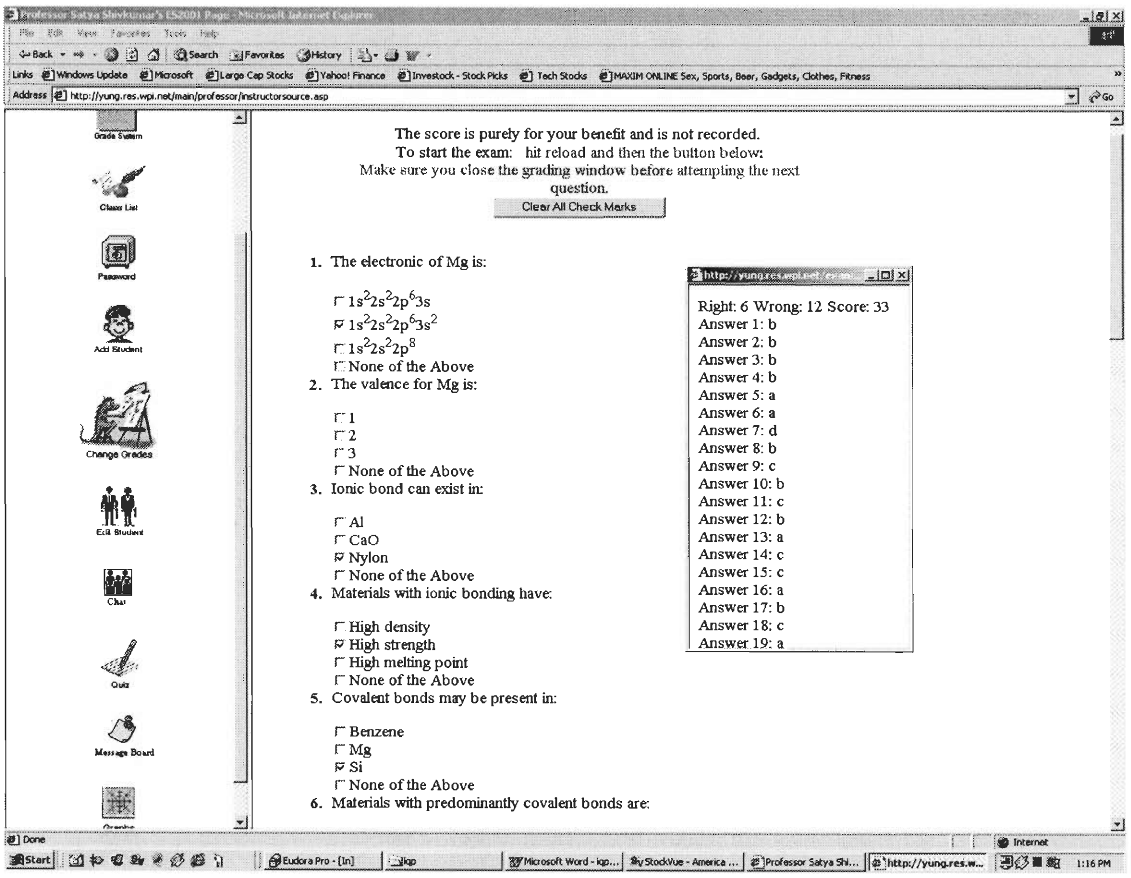

Secure Grading
Run login checker. (check user name and password again)
User name and password valid
Yes
No
Grab user name, quiz, exam, and homework grades
Load error web page
Display user name, quiz, exam, and home work grades in new web page

*Fig.16 - Flow chart of secure grading system.*

## *4.2.7 Quiz System*

The Quiz System was written in HTML with C code. A C code function was written to compare the answer chosen from the multiple choice or from the true or false question. Another function was implemented to keep track of the right/wrong answers. Finally, a further function takes care of score calculation for the user. Also written in C and HTML is the dynamically generated HTML answer key page. This is illustrated in the figure below.



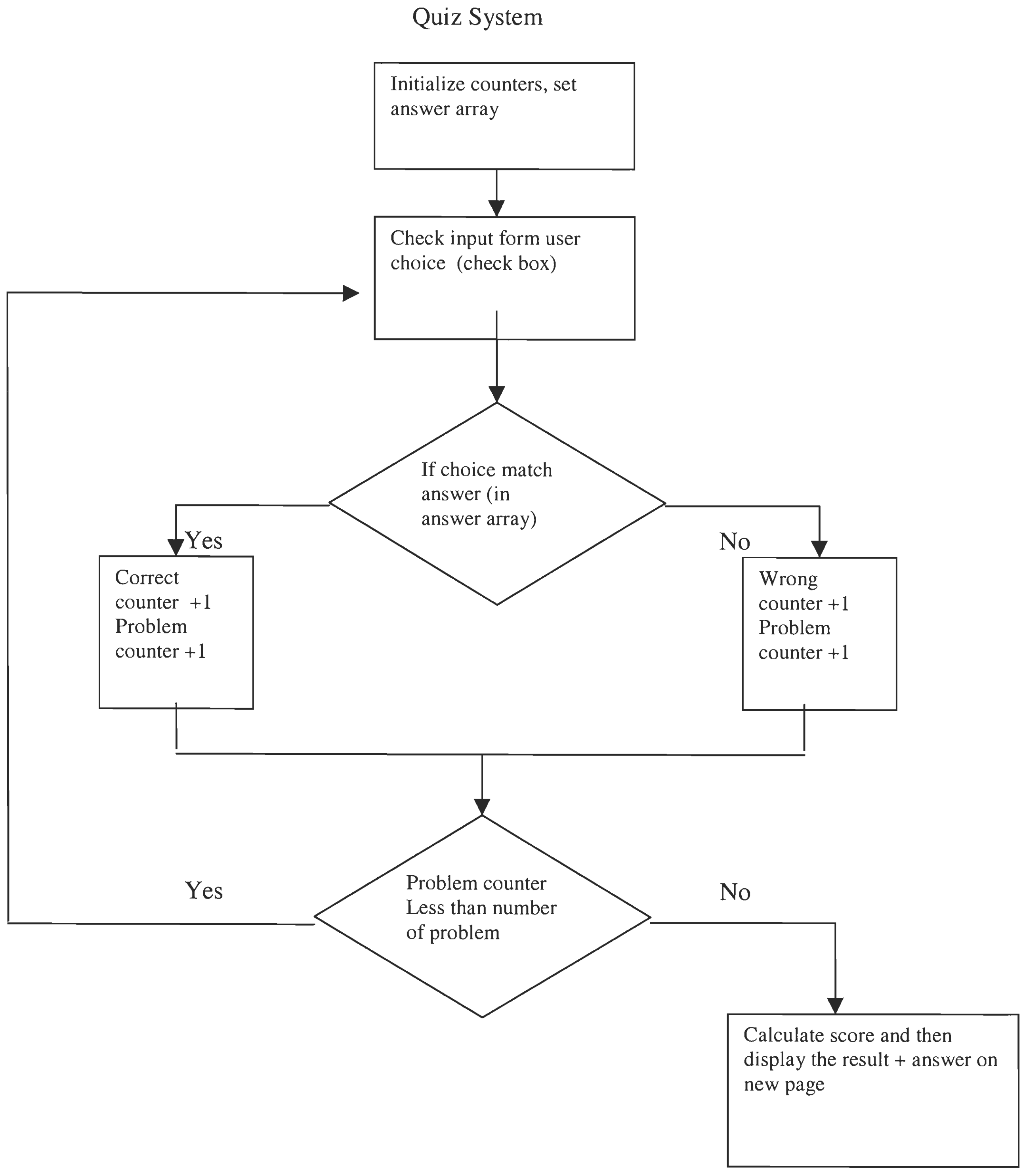
The score is purely for your benefit and is not recorded.
To start the exam: hit reload and then the button below:
Make sure you close the grading window before attempting the next question.

1. The electronic of Mg is:
   - $1s^2 2s^2 2p^6 3s$
   - [✓ $1s^2 2s^2 2p^6 3s^2$]
   - $1s^2 2s^2 2p^8$
   - None of the Above
2. The valence for Mg is:
   - 1
   - 2
   - 3
   - None of the Above
3. Ionic bond can exist in:
   - Al
   - CaO
   - [✓ Nylon]
   - None of the Above
4. Materials with ionic bonding have:
   - High density
   - [✓ High strength]
   - High melting point
   - None of the Above
5. Covalent bonds may be present in:
   - Benzene
   - Mg
   - [✓ Si]
   - None of the Above
6. Materials with predominantly covalent bonds are:

```
Right: 6 Wrong: 12 Score: 33
Answer 1: b
Answer 2: b
Answer 3: b
Answer 4: b
Answer 5: a
Answer 6: a
Answer 7: d
Answer 8: b
Answer 9: c
Answer 10: b
Answer 11: c
Answer 12: b
Answer 13: a
Answer 14: c
Answer 15: c
Answer 16: a
Answer 17: b
Answer 18: c
Answer 19: a
```

*Fig.17 - Automatic quiz system, display the answer and result after the student have finished the quiz.*

A flow chart showing the usage of the quiz system is shown here.

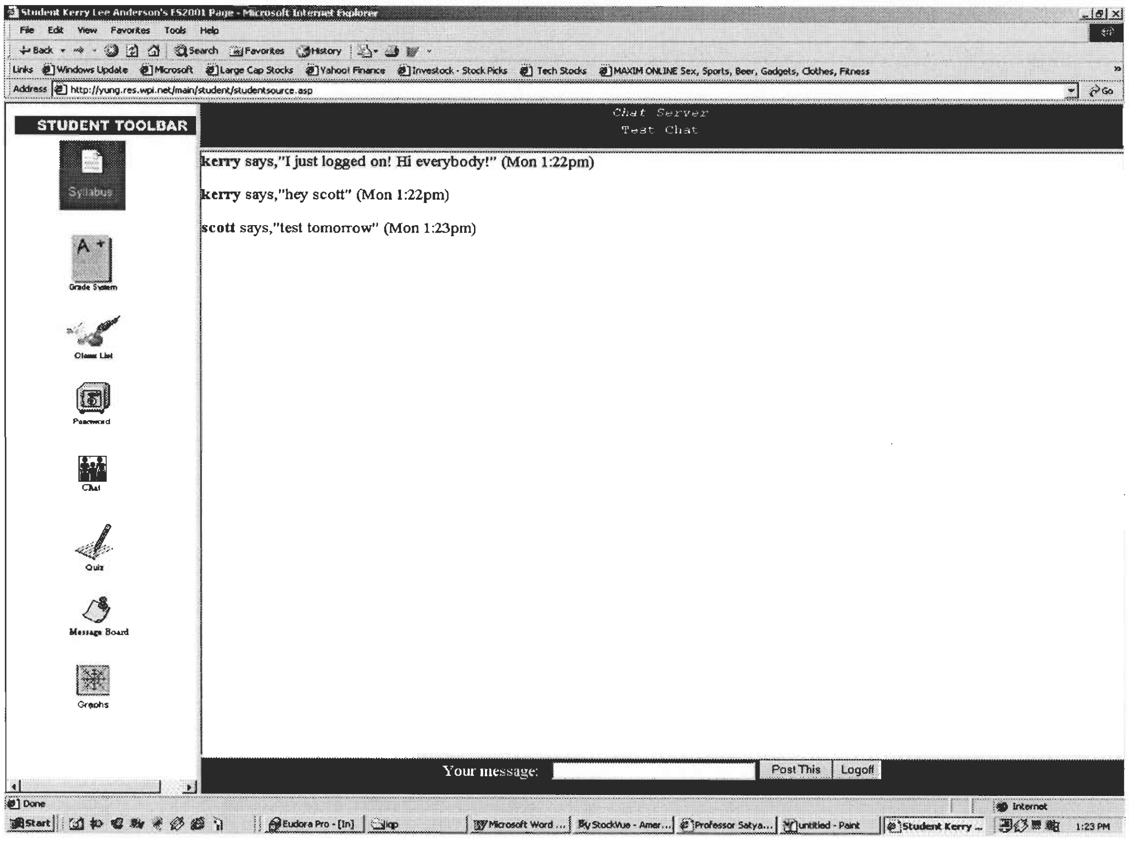


Quiz System
Initialize counters, set answer array
Check input form user choice (check box)
If choice match answer (in answer array)
Yes
No
Correct counter +1 Problem counter +1
Wrong counter +1 Problem counter +1
Problem counter Less than number of problem
Yes
No
Calculate score and then display the result + answer on new page

*Fig.18 - Flow chart of quiz system*

## *4.2.8 Chat Server*

The chat server shown below was written in Perl and HTML and is composed of three frames or pages of HTML. The Perl script is linked to the bottom frame, which is called upon when a user enters their login name. At this point, the user enters their information or chat room question. When the Perl script is executed it will update the main frame of the web page with the information that the user has entered.


Student Kerry Lee Anderson's FS2001 Page - Microsoft Internet Explorer
Address http://yung.res.wpi.net/main/student/studentsource.asp
STUDENT TOOLBAR
Syllabus
Grade System
Class List
Password
Chat
Quiz
Message Board
Graphs
Chat Server
Test Chat
Your message:
Post This
Logoff

**kerry** says,"I just logged on! Hi everybody!" (Mon 1:22pm)

**kerry** says,"hey scott" (Mon 1:22pm)

**scott** says,"test tomorrow" (Mon 1:23pm)

*Fig.19 - Display how the message with the user name plus the date and time the message was display.*

The process by which a user posts a message to the chat room is described in the flow chart below.

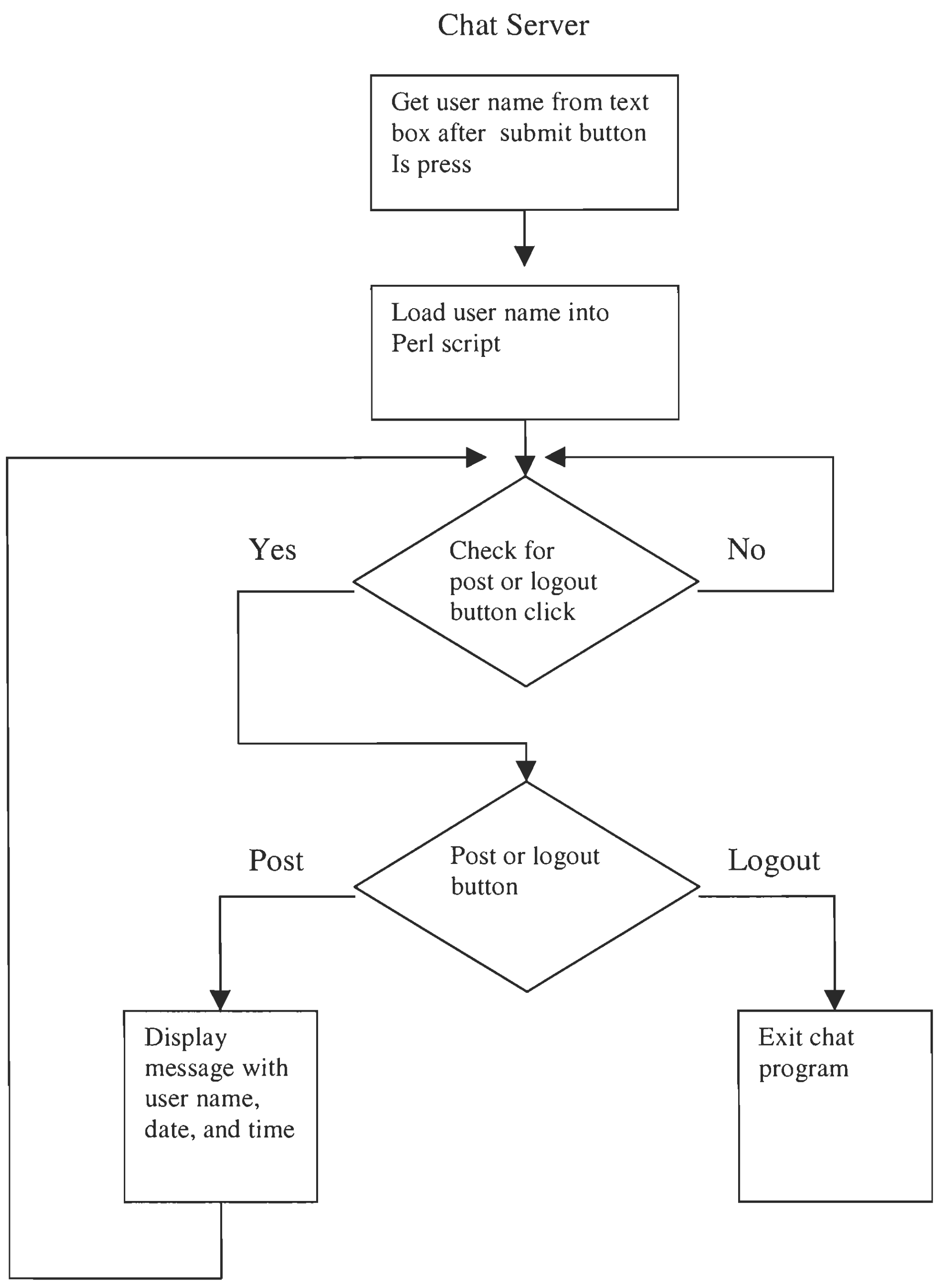

Chat Server
Get user name from text box after submit button Is press
Load user name into Perl script
Check for post or logout button click
Yes
No
Post or logout button
Post
Logout
Display message with user name, date, and time
Exit chat program

*Fig.20 - Chat server flow chart.*

## *4.2.9 Bulletin Board*

The bulletin board was developed in Perl with the help of an HTML text box and post and clear function. After the post button has been pressed, it will call on a Perl script which takes user input text and stores it in a data base. A function written in Perl will be used to load up all the stored messages from the database and will dynamically create an HTML file in which all the messages will be posted. A picture of how the user will write a message on the bulletin board is shown below.

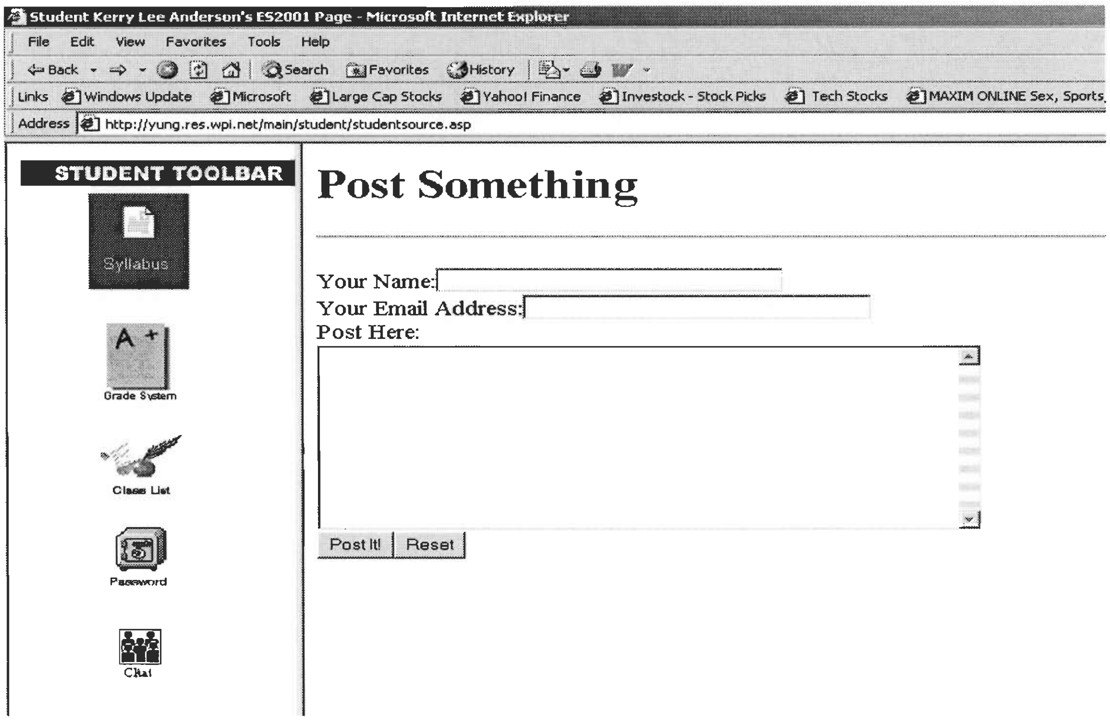

Student Kerry Lee Anderson's ES2001 Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Windows Update Microsoft Large Cap Stocks Yahoo! Finance Investock - Stock Picks Tech Stocks MAXIM ONLINE Sex, Sports
Address http://yung.res.wpi.net/main/student/studentsource.asp
STUDENT TOOLBAR
Syllabus
Grade System
Class List
Password
Chat
Post Something
Your Name:
Your Email Address:
Post Here:
Post It!
Reset

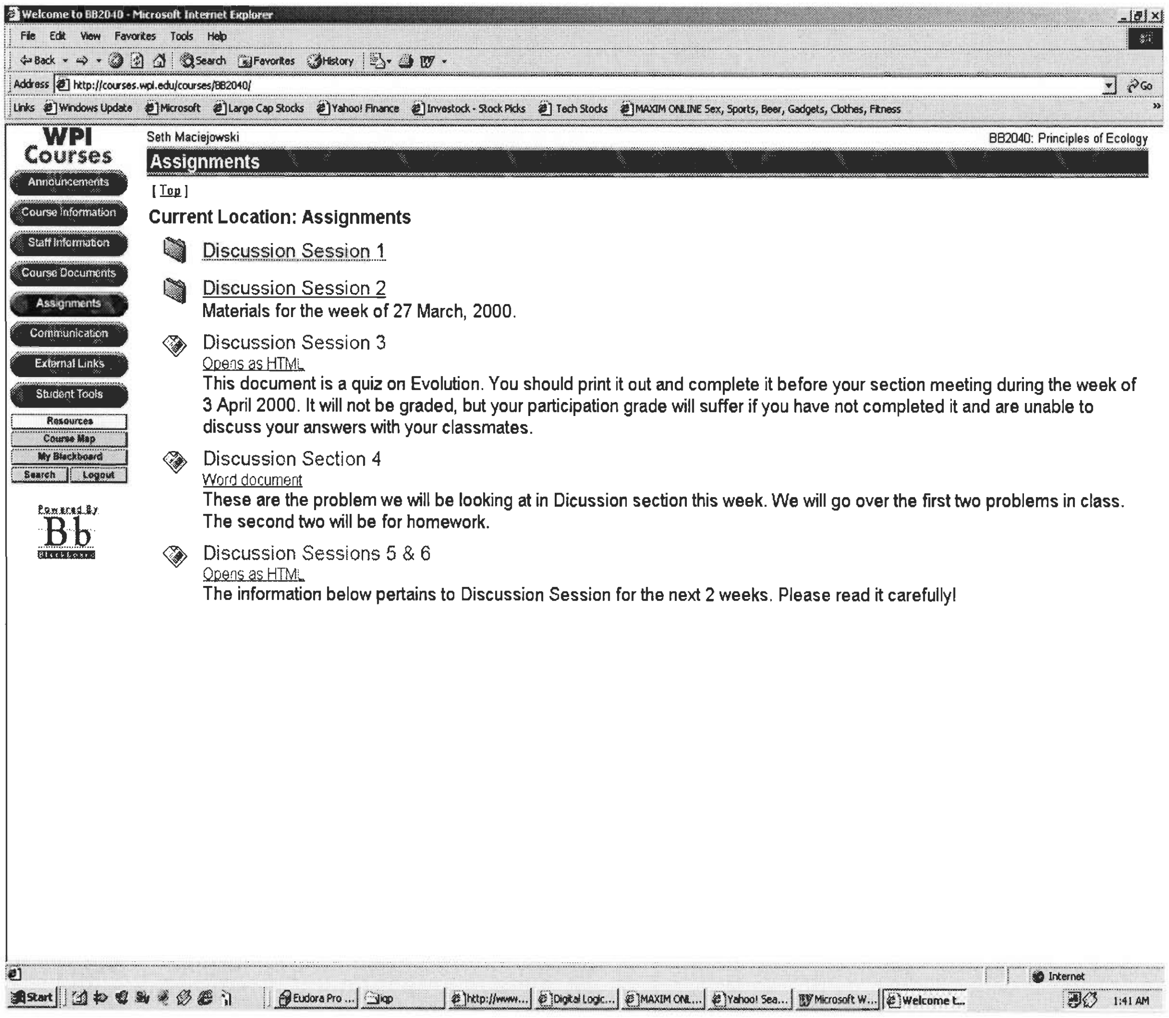
*Fig.21 - Posting message on the message board with the user name, e-mail address, and the message the want to post*

.

## 4.3 Black Board

After completing the student interactive modules using ASP, Perl and HTML, it was discovered that the main focus of the software had been achieved by software offered by a company known as Black Board. Black Board has generated a software package that acts as an interactive tool for students and professors in a given class. The Black Board interface offers many features, some of which have been integrated into the implemented version and some that this version does not offer. However, some features were developed that were not found in the Black Board interface.

The Black Board interface offers a number of features. On the Black Board interface one finds sections that outline course announcements, course information, staff information, and assignments.



Seth Maciejowski

BB2040: Principles of Ecology

**Assignments**

[ Top ]

**Current Location: Assignments**

Discussion Session 1

Discussion Session 2
Materials for the week of 27 March, 2000.

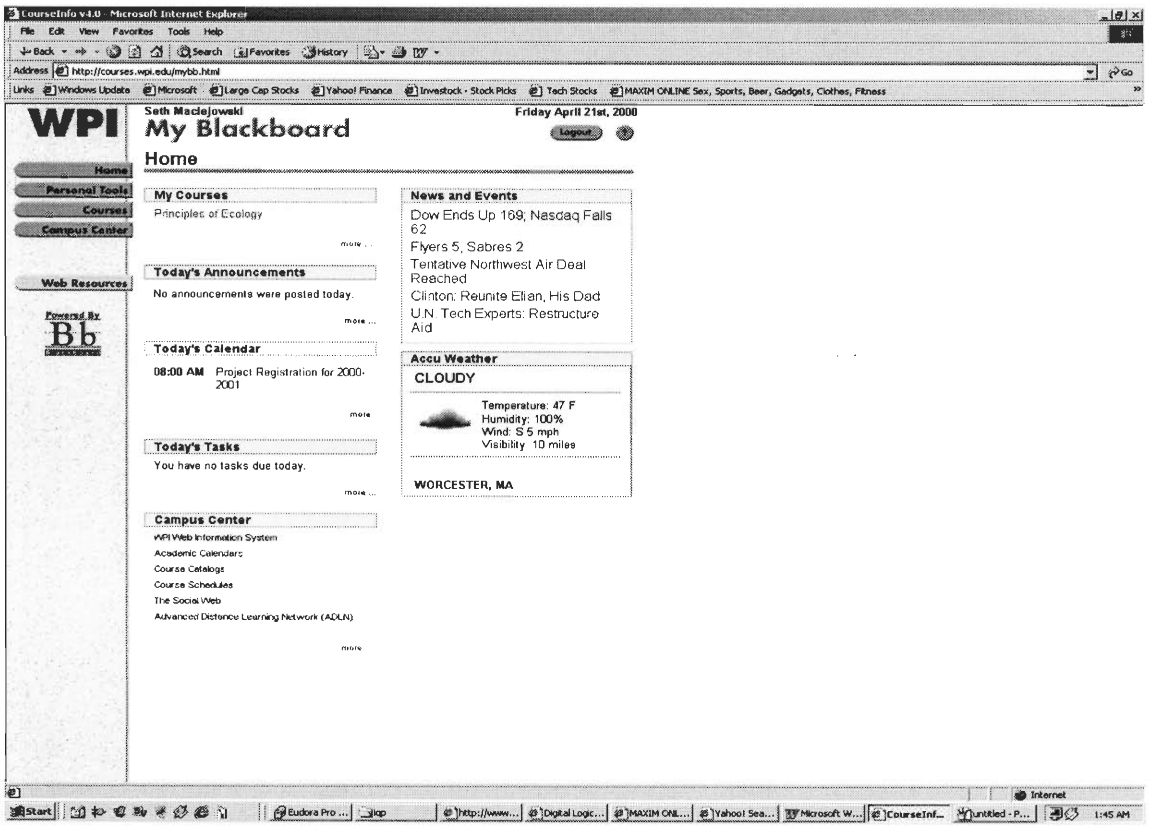
Discussion Session 3
Opens as HTML
This document is a quiz on Evolution. You should print it out and complete it before your section meeting during the week of 3 April 2000. It will not be graded, but your participation grade will suffer if you have not completed it and are unable to discuss your answers with your classmates.

Discussion Section 4
Word document
These are the problem we will be looking at in Dicussion section this week. We will go over the first two problems in class. The second two will be for homework.

Discussion Sessions 5 & 6
Opens as HTML
The information below pertains to Discussion Session for the next 2 weeks. Please read it carefully!

*Fig.22 – The main Black Board page with different modules user can choose from.*

The tools that Black Board offers students and professors include an external web links page, a calendar, a grade checker, an assignment drop box, a student information editor, a chat room, a student roster list, and a virtual classroom. A student may also download course documents using Black Board. Finally Black Board offers a general "My Blackboard" page that provides access to current news, weather and events on the WPI campus. The "My Blackboard" page provides a list of all courses that you are currently enrolled in and using Blackboard with. Finally, through the "My Blackboard" page, one receives direct links to the WPI main web page. These links include everything from residential services to academic advising.


CourseInfo v4.0 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://courses.wpi.edu/mybb.html
Links Windows Update Microsoft Large Cap Stocks Yahoo! Finance Investock - Stock Picks Tech Stocks MAXIM ONLINE Sex, Sports, Beer, Gadgets, Clothes, Fitness
WPI
Home
Personal Tools
Courses
Campus Center
Web Resources
Powered By Bb
Logout
Start
Internet
1:45 AM

Seth Maciejowski

My Blackboard

Friday April 21st, 2000

Home

**My Courses**

Principles of Ecology

more ...

**Today's Announcements**

No announcements were posted today.

more ...

**Today's Calendar**

08:00 AM Project Registration for 2000-2001

more

**Today's Tasks**

You have no tasks due today.

more ...

**Campus Center**

WPI Web Information System
Academic Calendars
Course Catalogs
Course Schedules
The Social Web
Advanced Distance Learning Network (ADLN)

more

**News and Events**

Dow Ends Up 169; Nasdaq Falls 62
Flyers 5, Sabres 2
Tentative Northwest Air Deal Reached
Clinton: Reunite Elian, His Dad
U.N. Tech Experts: Restructure Aid

**Accu Weather**

CLOUDY

Temperature: 47 F
Humidity: 100%
Wind: S 5 mph
Visibility: 10 miles

WORCESTER, MA

*Fig.23 – The "My Blackboard" page shows the weather and events of the day.*

The features that the Blackboard web interface offers that the implemented software does not offer include the student calendar, the ability to email to an entire group, and the ability to download documents. The software created for this project, however, offers some features that are not found on the Blackboard interface. These include an automated quiz correcting system, the 3d viewer described earlier in this document, a graphing calculator, and an experimental results data base in which students can add their experimental results online.

The focus of the Black Board software is to be a general interactive tool for any course offered at WPI. The software implemented for this project is specifically geared toward a materials science course. This means that the chat room center and the other special features described in this document are intended for a materials science audience.

## 4.4 Pros and Cons of Test Modules

Black board offers many features for users. This can be helpful, but it can be also a hindrance as well. The large amount of features could make blackboard difficult to use due to its complexity. Also, the titles for each feature in the user interface are not descriptive enough. This means that it is difficult to ascertain the usage of a given feature on the user interface by its button alone. The format of the blackboard user interface requires the user to have a good understanding of a web browser and other elements of the Internet. This is probably a necessity, however, and it could be covered in the help files. Finally, blackboard is intended as a general classroom tool whereas my web modules are specifically designed for a materials class. This means that utilities can be created that are specifically designed for this class.

Black board has many, many positive aspects. The features that it provides are extensive and informational. A student can gain access to any classroom data at a given time or immediately download documents and other course related data. The login for Black board is also secured, meaning that each user may have their own personal black board settings. The secure login allows blackboard to offers things such as a calendar scheduler and an address book.

Clearly, Black board fulfills the requirements for general courses, but a niche may still be found in developing web modules for specific courses.

## 4.5 Summary

The Internet has already greatly enhanced learning experiences for students in the past several years. With faster a connection, lectures could be taught and broadcasted using the Internet. With this feature, professors and the students can have real-time interactions. Professors may be able to give lectures from home. Since lectures can be given over the Internet, the number of students in a class will no longer be a limitation to class size. Since the lecture is broadcasted over the Internet, it can be recorded enabling students to reference the lecture at a later date.

Different methods of implementation can be accomplished through the use of different programming languages. Each language offers its own advantages and disadvantages. These advantages and disadvantages are summarized here:

- HTML
    - A web standard
    - Lacks interactivity ability
- Java

    - Offers good animation
    - Offers web security
    - Very difficult to code
- Perl
    - Easy to program (similar to C)
    - Good for internet data bases and client server interactions
    - No animation
- VRML
    - Offers 3d visualization
    - Little programming required
    - Large system requirements

Typical features found in a web based interactive module set include a login checker, secure grading system, chat server, quiz system, and bulletin board. These features offer instructors and students the ability to retrieve documents and ask questions online. The quiz system allows students the ability to take quizzes outside the classroom.

The features described here offer students and instructors the ability to communicate and receive help and instruction outside the classroom. Black Board offers many of the requisite features that are looked for in an interactive web based module set. Some of the advantages and disadvantages offered by Black Board are

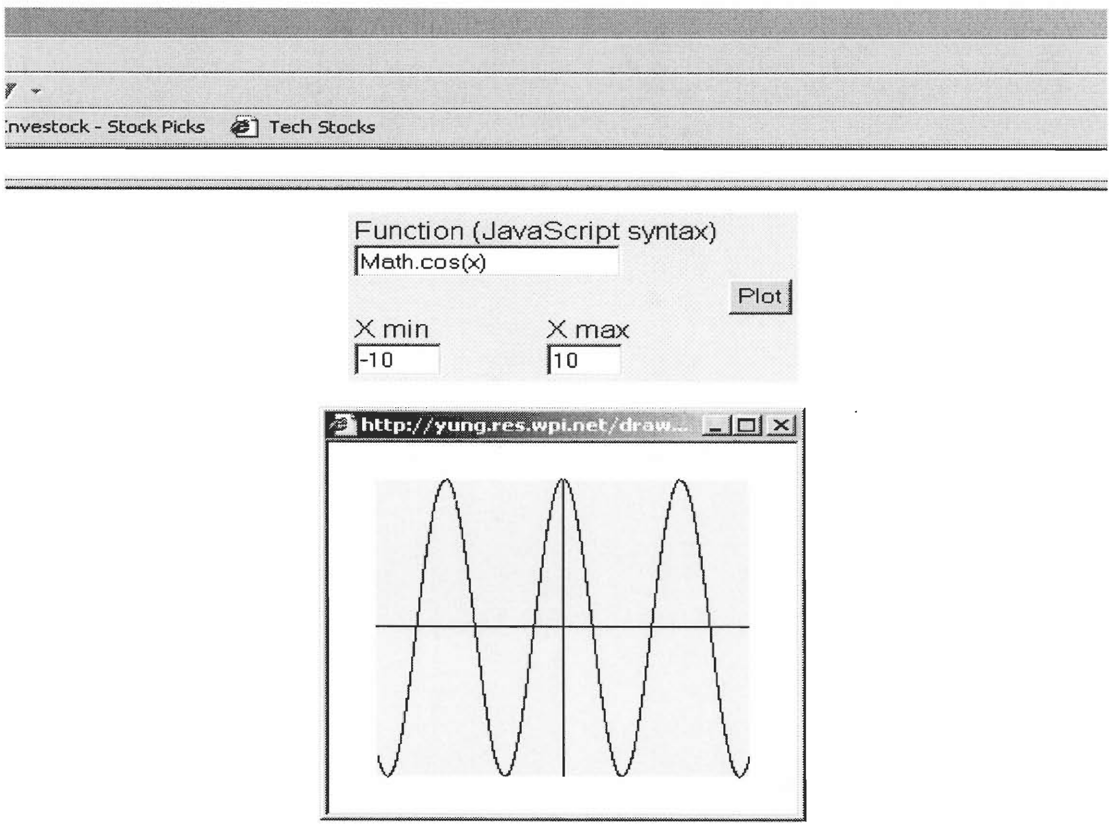
- Secure Login
- Calendar
- Grades
- Chat Rooms

Black Board lacks certain features that may prove useful in some courses such as the materials Science course here at WPI. These include:

- VRML
- Graphing calculator
- Online quiz system

## 5.0 Development of Interactive Graphical Modules

The graphing GUI (graphical user interface) was written using Java and HTML. The user enters a function and the limits of the function into the HTML post text box. The post text box then calls a number of Java math functions. After the inputs have been calculated though the calculation functions, another Java function is use to plot all the calculated points. After developing the graphing tools, another similar but more advanced web based graphing tool that would be more valuable for my project was found. Figures of both graphing GUI's are shown below.



Function (JavaScript syntax)
Math.cos(x)
X min
-10
X max
10

*Fig.24-Project development of graphing tool. Right*

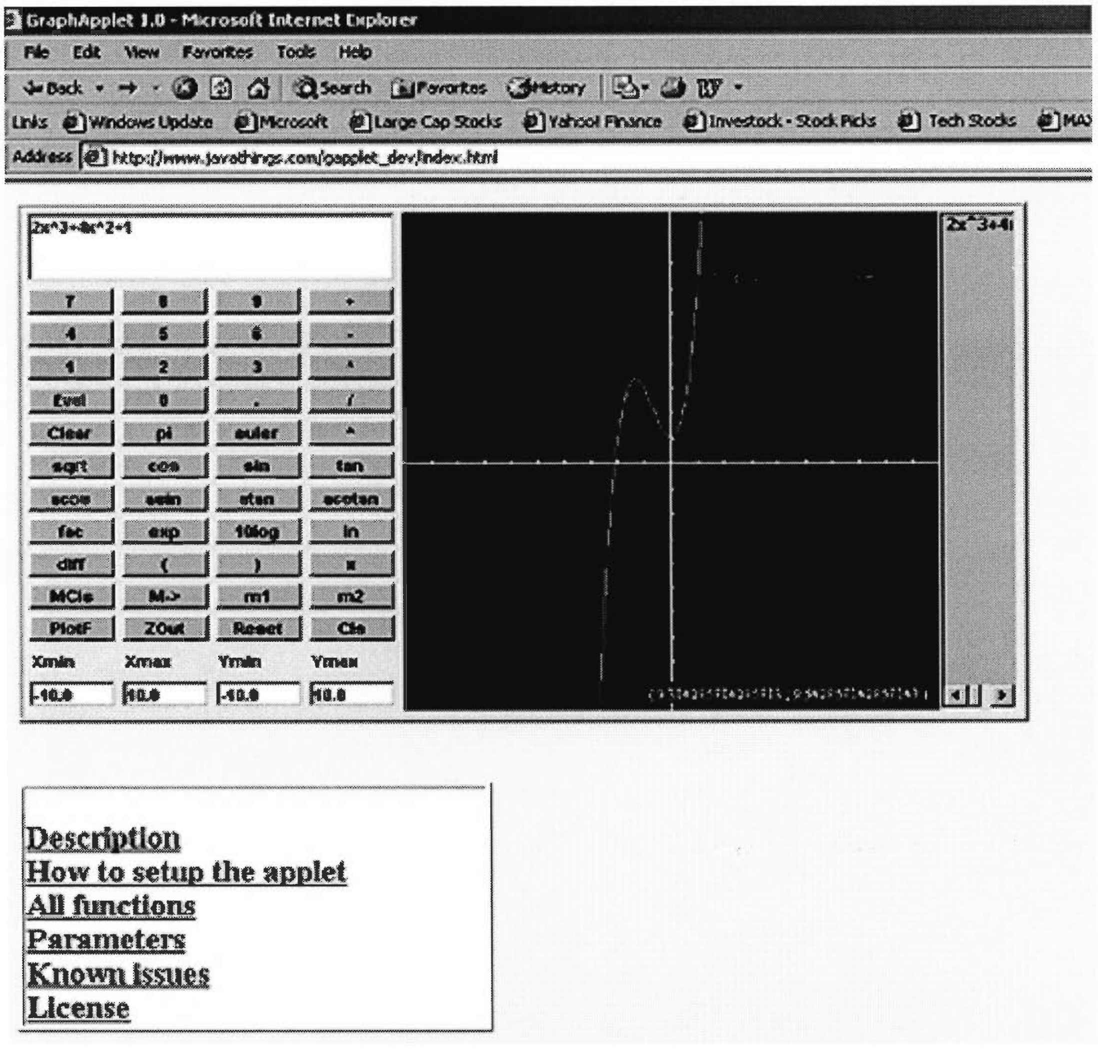


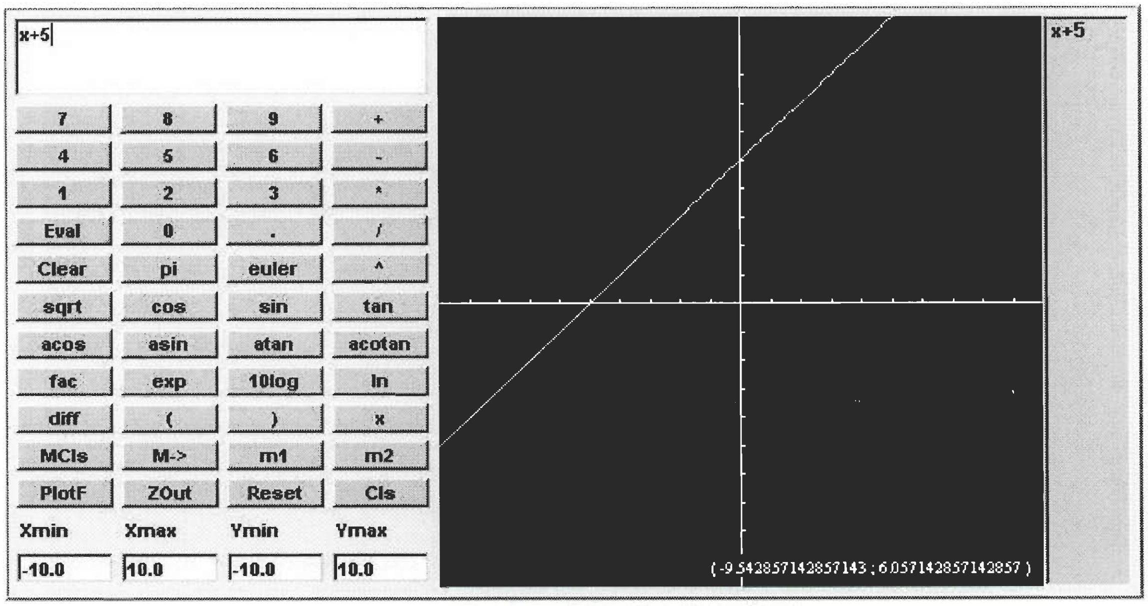
*Fig.25 – Graph Applet Developed by Patrik Lundin*

The graphing tool developed in this project has limited functionality in comparison with the GraphApplet that was developed by Patrik Lundin. As a result Patrik Lundin's GraphApplet was chosen and altered to meet the project expectations. The GraphApplet has the common operators and functions provided by any scientific calculator as well as function graphing capabilities. The GraphApplet supports juxtaposition which means you can write your expressions just as you would write them on a piece of paper, an example of this being xcosx or 5x^3-6x. The GraphApplet also has support for a symbolic differentiation using the diff function. The applet is used in the same fashion as a calculator. A user can click on the buttons of the applet or use the keyboard to form the expression they want in a text box. After entering the equation the enter button is clicked and a graph is produced. If the Eval button is used, the equation will be evaluated for a given set of values. A function that will be graphed has to take the form y=f(x) to facilitate graphing. To zoom in the plotted area you can use the mouse to generate a zoom window over the area that you want to zoom in on. To zoom out the Zout button on the applet is used. The user can also set boundaries so that they can view the graphed function in any quadrant. A tablet below shows the value to set Xmin, Xmax, Ymin, and Ymax to show each quadrant.

*Table.6 - GraphApplet quadrants, values for Xmin, Xmax, Ymin, and Ymax.*

| | Xmin | Xmax | Ymin | Ymax |
|---|---|---|---|---|
| Quadrant 1 | Num >= 0 | Num > Xmin | Num >= 0 | Num > Ymin |
| Quadrant 2 | Num < Xmax | Num <= 0 | Num >= 0 | Num > Ymin |
| Quadrant 3 | Num < Xmax | Num <= 0 | Num < Ymax | Num <= 0 |
| Quadrant 4 | Num >= 0 | Num > Xmin | Num < Ymax | Num <= 0 |
| Quadrant 1+2 | Num < 0 | Num > 0 | Num >= 0 | Num > Ymin |
| Quadrant 2+3 | Num < Xmax | Num <= 0 | Num < 0 | Num > 0 |
| Quadrant 3+4 | Num < 0 | Num > 0 | Num < Ymax | Num <= 0 |
| Quadrant 1+4 | Num >= 0 | Num > Xmin | Num >= 0 | Num > Ymin |

Shown below are some pictures of the feature that the Graph Applet have.



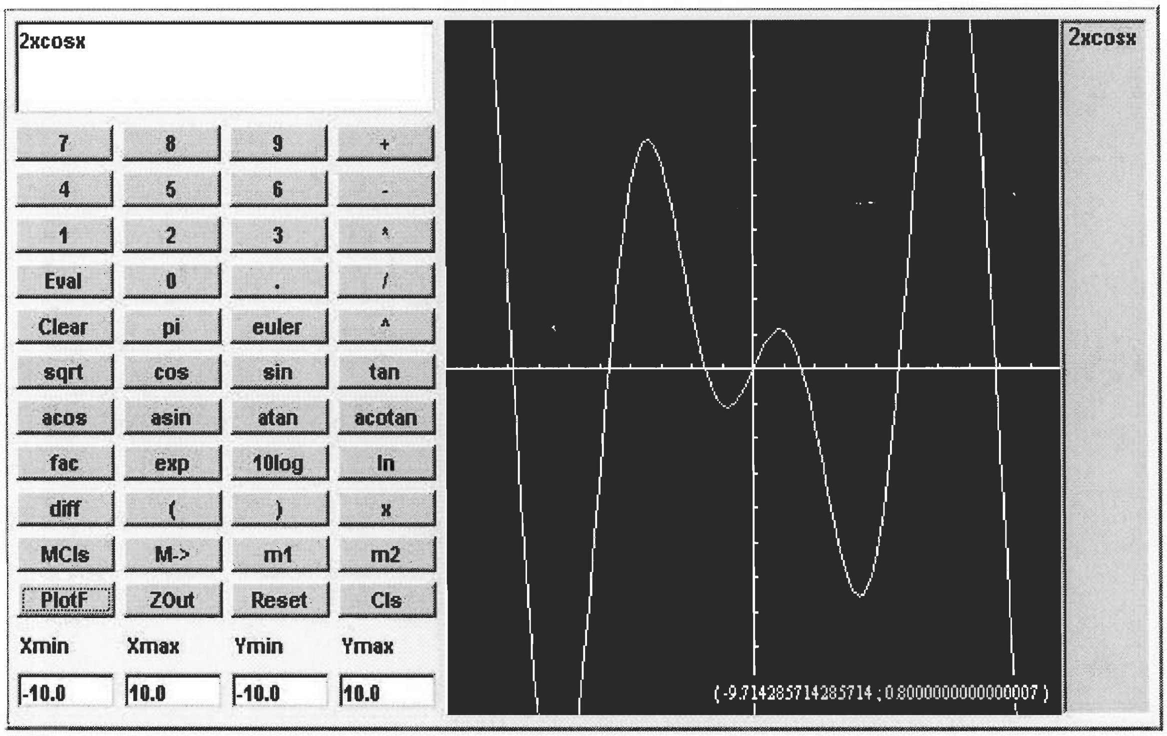
x+5

| Xmin | Xmax | Ymin | Ymax |
|---|---|---|---|
| -10.0 | 10.0 | -10.0 | 10.0 |

( -9.542857142857143 ; 6.057142857142857 )

*Fig.26 - Graph of the function X+5.*

The Graph Applet can plot the cosine, sine, and tangent function



2xcosx

| Xmin | Xmax | Ymin | Ymax |
|---|---|---|---|
| -10.0 | 10.0 | -10.0 | 10.0 |

( -9.714285714285714 ; 0.8000000000000007 )

*Fig.27 - Graph of the function 2xcosx.*

The GraphApplet can plot multiple graphs on the applet screen. Multiple functions can not be enter in at the same time but the applet will plot a new function to the same graph of a previous plot. This allows multiple graphed function to be displayed on the same screen for comparison. Shown in figure??? Are multiple functions graphed on the same screen. The list of functions that the GraphApplet has is in appendix ???.

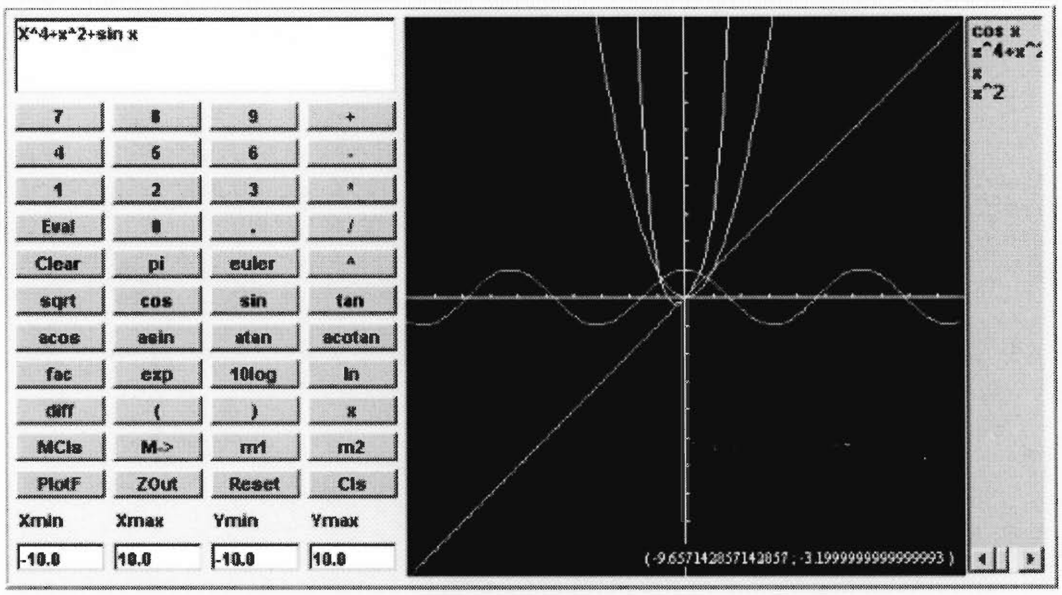


*Fig.28 - GraphApplet, Multiple functions are plotted on the same screen.*

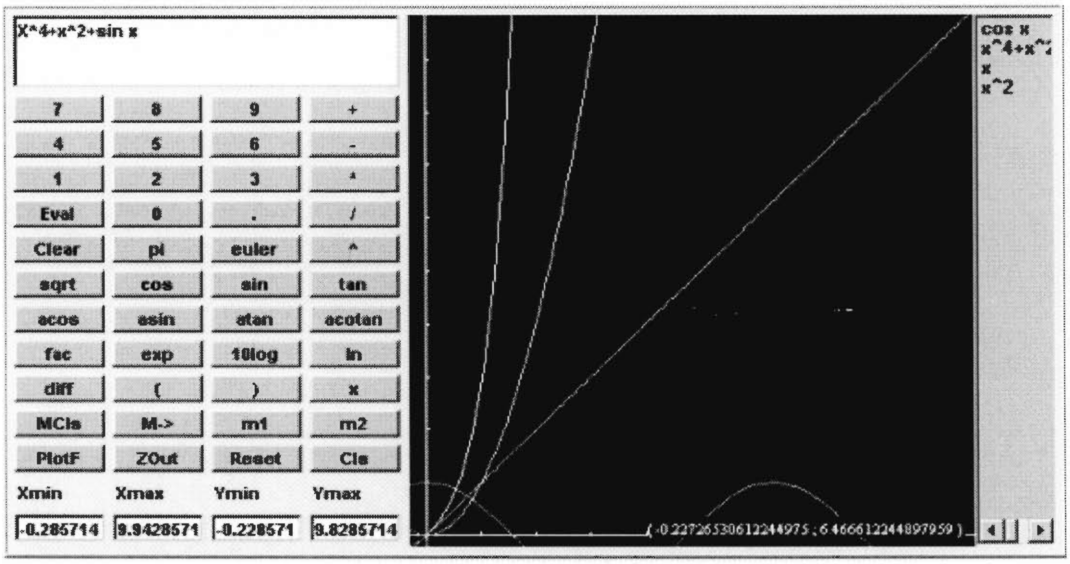
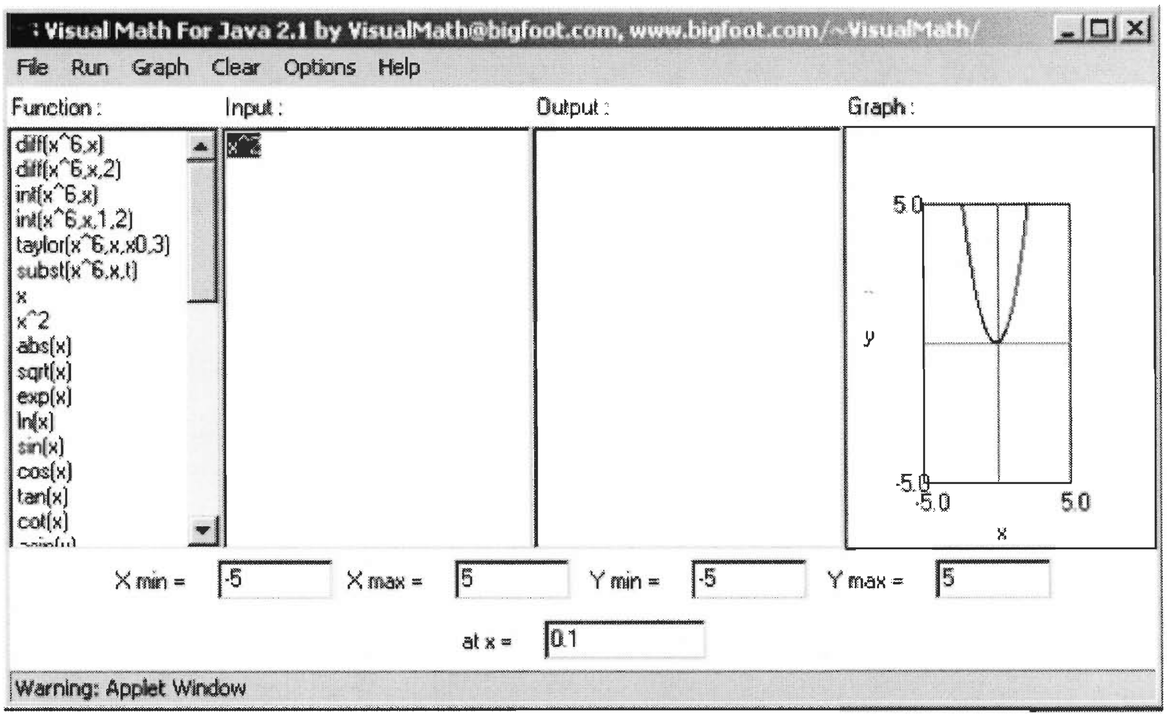


*Fig.29 - GraphApplet, Zoom in version of the Multiple function graph.*

Other Interactive Graphical Modules with greater functionality are available, but they do not have the simple user interface offered by GraphApplet. The image below shows another graphing utility offered by Visualmath that offers more features, but is difficult to use.



X min = -5 X max = 5 Y min = -5 Y max = 5

at x = 0.1

*Fig.30 – VisualMath for Java 2.1*

GraphApplet requires little time to learn for the amount of functionality it offers in comparison with more advanced utilities.

## 5.1 Summary

An online calculation tool is important, particularly for a science or math class because it may be used in conjunction with an online resource for long distance learning. In this project, three different calculation tools were analyzed. Each had their own advantages and disadvantages. These pros and cons are listed below.

- Hand Coded calculation tool
    - Simple to use
    - May be tailored to certain class types
    - Limited functionality
- Graph applet
    - Simple to use
    - Offers a large range of graphing capability
    - Limited functionality

The advantages to using an online calculator versus a physical calculator is that some students may not have a calculator possessing the requisite functionality for certain courses. The functionality required for certain courses may be coded into a specific interactive module and used by the entire class. Also, if an online calculator is used, the figures that each student gets will be in agreement and not off by small amounts depending upon the calculator used.

## 6.0 Development of Modules for Materials Science

The 3-D graphing simulator was written in pro-engineer. Layouts were created using various functions and graphical interfaces that were provided by the pro-engineer software. There were multiple shapes chosen to create the 3-d image. After the right image was created, it was exported as a VRML file type. To view this VRML a user has to upgrade their Internet browser with the Cosmo viewer. A picture of one of the 3-d models is shown below.

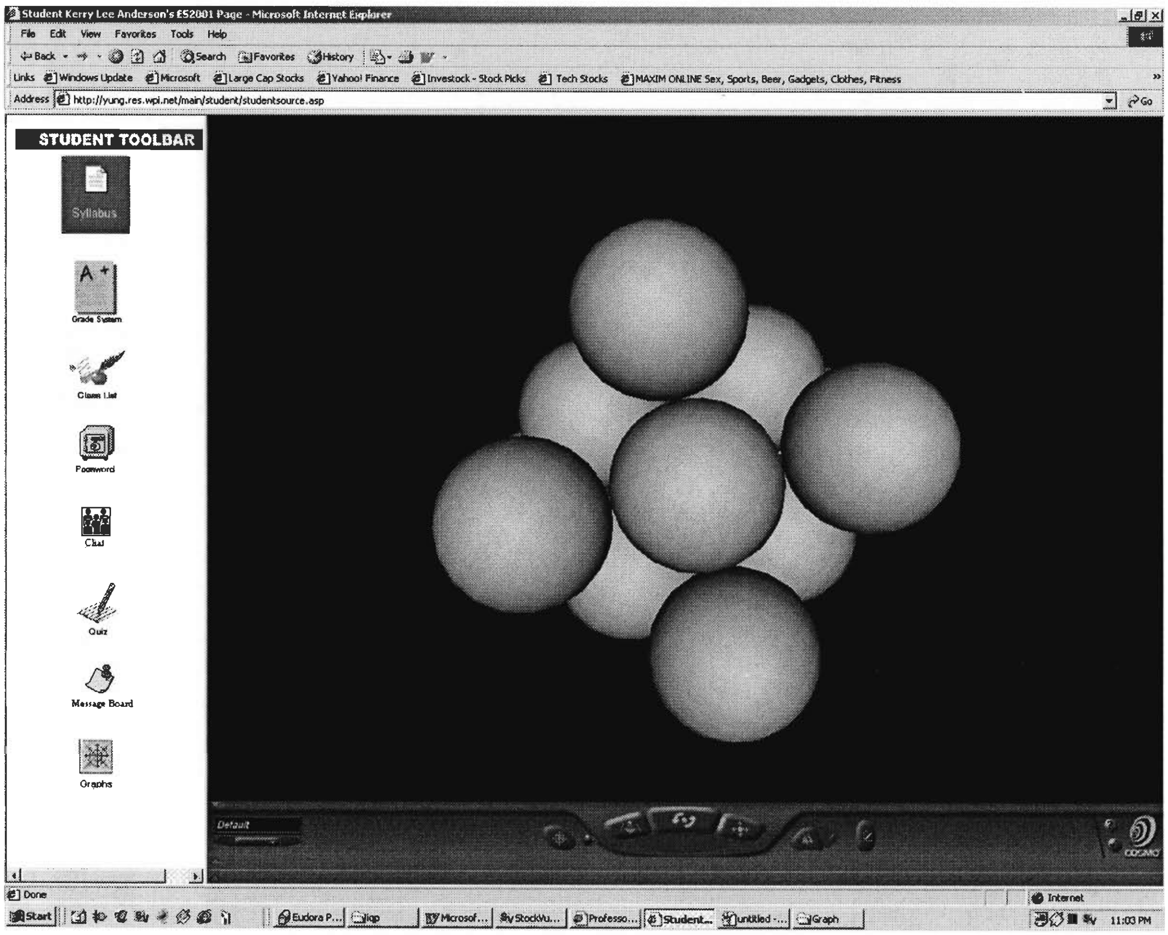

Student Kerry Lee Anderson's ES2001 Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Windows Update Microsoft Large Cap Stocks Yahoo! Finance Investock - Stock Picks Tech Stocks MAXIM ONLINE Sex, Sports, Beer, Gadgets, Clothes, Fitness
Address http://yung.res.wpi.net/main/student/studentsource.asp
Go
STUDENT TOOLBAR
Syllabus
Grade System
Class List
Password
Chat
Quiz
Message Board
Graphs
Default
COSMO
Done
Internet
Start
Eudora P...
Microsof...
StockVu...
Professo...
Student...
untitled -...
Graph
11:03 PM

*Fig.31 - 3-D picture of a face center cubic structure viewing with the Cosmo 3-D player.*

These 3-d models provided are of common molecular lattice structures. Due to the complexity in atomic arrangement and positioning in these structures, 2-D paper drawings and descriptions are often inadequate. 3-d physical models are often much better for teaching and learning these structures, but their size, cost, and limited availability makes them difficult to distribute to a large group of students. Here is where the 3-d VRML models come in. They provide the 3-d capabilities of a physical model, as well as the 2-d mass distribution qualities over the Internet.

The way that the VRML lattice models provide 3-d capabilities is that the viewer of a model may manipulate the model as if it was a physical object. This means the viewer provides rotate, zoom, and pan commands in all three axes. The user can rotate the structure to a particular face, pan to center on an atom on that face, and then zoom in to view adjacent touching atoms. In addition to these commands, there is a walkthough command that allows the user to "walk" through the molecular structure as if they are walking through a museum exhibit.

Furthermore, other advantage of using VRML is that it allows object linking and paths. Object linking is associating an object in VRML, such as an atom (a sphere), with a text label. This will cause the label to be shown when the mouse is over the particular object. HTTP links may also be embedded into objects so that clicking on an object will bring up a web page. This is extremely useful when one would like to show additional details about an object. Finally, paths are points in the 3-d space that the object occupies that the model will automatically go to. The use would press the Follow path button and the object would animate and go to these points in the 3-d space. This allows the educator to show important details about an object such as faces, planes, and bond angles to any use at any time since these paths are saved in the 3-d model.

A graphical 3-d model was developed to study crystal structure in materials. This model enables rotation and examination of the structure from various angles. It provides a better understanding of the structure. In the future, the model can be used to provide the following:

- The capability of removing an atom and examining the residue structure.
- More complex structures can be studied better.
- Voids in the structure can be viewed accurately.
- The ability to detach and combine atoms to form molecules.

## 6.1 VRML Features Summary

- Like physical model but easily distributable.
- More functionality than physical model through walk through and preset views
- "Paths" enable a virtual demonstration
- Object linking allows details to be linked to designated objects.

| VRML | Handout |
|---|---|
| 3-d, Like actual molecular structure | 2-d |
| Electronically distributed, available 24 hrs/day | Only available in class |
| Allows virtual demonstration through "paths" | No demonstration |
| Object linking allows details to be linked to specific atoms and molecules through clicking. | Restricted to traditional lines which can be confusing in a complex structure. |
| VRML viewer requires learning. | Traditional learning tool, just read. |

*SHADED = Advantage

Table.7 VRML vs. Handout.

# 7.0 Summary

The intent of this project was to analyze the interactive web modules offered by different schools and then make decisions about what would be the most useful, beneficial implementation for WPI.

The interactive course web sites for many other schools were evaluated on a 1 to 10 scale, based upon the features offered, ease of use accessibility and security. The features of each school were recorded to determine what is generally found on an interactive web site. Typically the following items are found on a schools interactive course web site:

- Secure login
- Chat server
- Bulletin board
- Announcements
- Calender
- Quizzes
- E-Mail

The interactive module developed for this project was completed with a specific course in mind, namely Materials Science. As a result, specific features, such as the VRML viewer were added to enhance the web page for this specific course. Many of the previously mentioned items are included in the interactive module set as they are useful in any course.

The graphing module used for this project was developed by an outside source. This source was selected over a hand coded graphing module because it offered simplicity of use and a relatively large function set over that offered by the hand coded version. The graphing module is useful for experimentation where variables may be entered and graphed for different intervals. The module is developed such that it allows for easy future expansion for use in specific course web sites or projects. Other implementations were looked at that offered a larger function set, but these were disregarded due to a difficult to use user interface.

The 3d module specifically designed for materials science is useful because it offers an enhanced method in which students can view 3d structures. The only disadvantages to this method of viewing 3d models over the traditional paper representation is that the software has large system requirements and also requires a little learning to understand how to use the software. The software allows a student to rotate and view a model from different angles and zoom depths using the web interface. This is a great improvement over the traditional method which offers only a 2d representation of a 3d model. In future implementations of this module you will be able to separate molecules and combine them to make more complex molecules.

The system in use at WPI is the Black Board system. This system offers a great deal of functionality including chat rooms, calendars, secure grading system, secure login, bulletin boards, news, and announcements. This system was evaluated to determine how useful it was for the WPI campus as a whole. The Black Board systems appears to offer a great deal of functionality that improves the classroom experience by adding help outside of the classroom that is easily accessible via the internet. Also, students find it easier to prepare for tests and exams by using documentation provided on the web site. The calendar is useful to indicate to students when important dates are approaching.

Black Board is used extensively here at WPI. On the courses web page we see 139 registered classes. This definitely indicates that the Blackboard software has been adopted with open arms by WPI. An indication of how intensively the software is being used by WPI can be seen in the example of a typical class such as Ecology BB2040. On that web module we see that the instructor and teaching assistants use the web page to convey class information, important documents, and announcements to the class. Also, a student may retrieve their grade or talk to their fellow classmates using an interactive chat module. Due to the overwhelming success of Black Board, this project recommends its use in Material Science over the developed Interactive Web Modules.

## 8.0 References

Francis, Brian, Kaufman, John, Llibre, Jaun T, Sussman, David, Ullman, Chris, **Beginning Active Server Pages 2.0**,
Wrox Press, Ltd: Birminghan, UK, 1998

Lemay, Laura, and Cadenhead Rogers, **Java 1.2 in 21 Days**.
Sams Publishing: Indianapolis, Indiana, 1998

*[WWW] http://www-mtl.mit.edu/Courses/6.095/spring-00/,* **6.095 / 2.995 - Spring 2000 Home Page***,* Massachusetts Institute of Technology, Cambridge, MA, 2000

*[WWW] http://webct.uconn.edu:8900/webct/public/show_courses.pl?947611874*, **WebCT Course Listing**, University of Connecticut, Hartford, CT, 2000

*[WWW] http://www.ece.unh.edu/courses/ee543_s00/ee543.htm*, **EE 543**, University of New Hampshire, Durham, NH, 2000

*[WWW] http://www.bu.edu/mfg/icv/.* **BU College of Engineering Course Page**, Boston University, Boston, MA, 2000

*[WWW] http://www.cs.cornell.edu/cucs/courses_degreeprogs/index.htm*, **Cornell Computer Science - Courses and Degree Programs**, Cornell, Ithaca, NY, 2000

*[WWW] http://www.umass.edu/webct/,* **The University of Massachusetts Amherst: WebCT Homepage: Main**, University of Massachusetts, Amherst, MA 2000

*[WWW] http://www.kn.pacbell.com/wired/vidconf/,* **Videoconferencing for Learning**, Pacific Bell, 2000

*[WWW] http://www.math.nyu.edu/research/ustilov/teach/,* **Calculus II Homepage**, New York University, New York, NY, 2000

# Appendix 1 – Calculator Controls

**Eval** - Evaluates an expression, the calculator first tries to evaluate the expression, if it fails then it tries to draw the expression.

**Clear** - Clears the input area.

**PlotF** - Plots a function of type y=f(x)

**ZOut** - Zooms out the current area with a factor of x2.

To Zoom into an area, select the area with the mouse and release.

**Reset** - Reset the bounds to the original x:[-10,10] y:[-10,10]

**Cls** - Clears the whole drawing area and all functions.

To clear an individual function double click on the function in the list to the right of the applet.

**M->** - Enters an expression or a value into memory, first click this button and then chose either m1 or m2

**MCls** - Clears the memory, first click this button and then on either m1 or m2 to clear that memory.

Note that memory will be gone if you leave the page with the applet and then return at a later point.

**m1** - Memory 1

**m2** - Memory 2

Operators and Functions

**sqrt** - sqrt of a value or expression.

**sin** - sin of a value or expression.

**cos** - cos of a value or expression.

**tan** - tan of a value or expression.

**atan** - atan of a value or expression.

**acos** - acos of a value or expression.

**asin** - asin of a value or expression.

**acotan** - acotan of a value or expression.

**exp** - The constant euler raised to a value or expression.

**ln** - The natural logarithm.

**10log** - The 10 logarithm of a value or expression.

**fac** - The faculty of a value or expression.

**diff** - Symbolically differentiates an expression of one variable

**pi** - The constant PI

**euler** - The Euler constant, base for the natural logarithm.

**+** - Addition operator.

**-** - Subtraction operator.

***** - Multiplication operator.

**/** - Division operator.

**^** - Raised to operator.

The calculator also supports the following functions when entered through the keyboard:

**sinh** - Hyperbolic sine

**cosh** - Hyperbolic cosine

**tanh** - Hyperbolic tangens

**abs** - Absolute value of value or expression.

**ceil** - Returns the smallest (closest to negative infinity) value that is not less than the argument and is equal to a mathematical integer.

**floor** - Returns the largest (closest to positive infinity) value that is not greater than the argument and is equal to a mathematical integer.

**sfac** - Semifaculty of value or expression.

**round** - rounds the argument to the closest mathematical integer.

Example: round( 3.4 ) = 3.0, round( 4.99 ) = 5.0 etc...

**fpart** - returns the decimalvalue of its argument

[base]log(..) - any logaritm. Example: 10log(10) = 1.0, exp(1)log(exp(1)) = 1.0

% - Modulo.

**==** - Equal, returns 1.0 if it's arguments are equal or 0.0 othervise.

**!=** - Not Equal, returns 1.0 if it's arguments are not equal or 0.0 othervise.

**&&** - And, returns 1.0 if both arguments evaluates to 1.0, or 0.0 othervise.

|| - Or, returns 1.0 if any of it's arguments evaluates to 1.0, or 0.0 othervise.

**>** - Larger than, returns 1.0 if the value of the argument to the left is larger than the value of the argument to the right, or 0.0 othervise.

< - Less than, returns 1.0 if the value of the argument to the left is less than the value of the argument to the right, or 0.0 othervise.

>= - Larger than or equal to, returns 1.0 if the value of the argument to the left is larger than or equal to the value of the argument to the right, or 0.0 othervise.

<= - Less than or equal to, returns 1.0 if the value of the argument to the left is less than or equal to the value of the argument to the right, or 0.0 othervise.

! - Not, returns 0.0 if it's argument evaluates to 1.0 and returns 1.0 if it's argument evaluates to anything other than 1.0

## Appendix 2 – Main Page HTML Source Code

```
<html>

<head>
<title>IQP</title>

</head>

<body TEXT="#000000" BGCOLOR="#FFFFFF" LINK="#ffffff" VLINK="#FFFFFF"
ALINK="#ffffff">
<font SIZE="+2" COLOR="#008080" FACE="Dom Casual"><b>


<font size="7"><b>

<p align="center">Interactive Web-Based Modules </p>
</b></font>

<p align="center"><br>
Please select your appropriate link to log in.</b></font> <br>
&nbsp; <br>
&nbsp; <!--webbot bot="ImageMap" rectangle="(0,20) (150,90) instructorlogin.htm"
rectangle="(320,0) (465,55) studentlogin.htm" rectangle="(90,245) (270,300)
adminlogin.htm" SRC="welcome.gif" startspan -->
<MAP NAME="FrontPageMap"><AREA SHAPE="RECT" COORDS="0, 20, 150, 90"
HREF="/main/instructorlogin.asp">
<AREA SHAPE="RECT" COORDS="320, 0, 465, 55"
HREF="/main/studentlogin.asp">
<AREA SHAPE="RECT" COORDS="90, 245, 270, 300"
HREF="/main/adminlogin.htm"></MAP>
<a href="_vti_bin/shtml.dll/index.htm/map"><img ismap usemap="#FrontPageMap"
height="312" src="welcome.gif" width="473">
</a><!--webbot bot="ImageMap" endspan i-checksum="29810" --> </p>
<p><br>
<br>

</body>

</html>
```

## Appendix 3 – Professor's Login Source Code

```
<!-- #include file="adovbs.inc" -->
<html>
<title>Instructor Login Page</title>
<form action="professor/instructorsource.asp" method="POST"><center>
<select name="CLASSID" size="1">
<%      dim objRS
        dim objConn
        dim strSQL

        strSQL = "SELECT ClassID, TeacherName FROM ClassIndex"
        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        Set objRS = Server.CreateObject("ADODB.Recordset")

        objRS.Open strSQL, objConn, adOpenKeySet, adLockReadOnly, adCmdText

        Do While Not objRS.EOF
                Response.Write "<option value = "%>"<%
                Response.Write objRS("ClassID")%>"<%
                Response.Write ">"
                Response.Write objRS("ClassID")
                Response.Write " - "
                Response.Write objRS("TeacherName")
                Response.Write "</option>"
                objRS.MoveNext
        Loop
        objRS.Close
        set objRS = Nothing
        objConn.close
        set objConn = Nothing
%>
</select>
<hr>
Login Name: &nbsp &nbsp &nbsp <input type="text" size=8 maxsize=8
name="LOGIN"><br>
Password: &nbsp &nbsp &nbsp <input type="password" size=8 maxsize=8
name="PASSWORD"><p>
<input type="submit" value="Log-In"></p><hr>
</form>
</html>
```

## Appendix 4 – Student's Login Source Code

```
<!-- #include file="adovbs.inc" -->
<html>
<title>Student Login Page</title>
<form action="student/studentsource.asp" method="POST"><center>
<select name="CLASSID" size="1">
<%
        Dim objRS
        Dim objConn
        Dim strSQL

        strSQL = "SELECT ClassID, TeacherName FROM ClassIndex"
        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        Set objRS = Server.CreateObject("ADODB.Recordset")

        objRS.Open strSQL, objConn, adOpenKeySet, adLockReadOnly, adCmdText

        Do While Not objRS.EOF
                Response.Write "<option value = "%>"<%
                Response.Write objRS("ClassID")%>"<%
                Response.Write ">"
                Response.Write objRS("ClassID")
                Response.Write " - "
                Response.Write objRS("TeacherName")
                Response.Write "</option>"
                objRS.MoveNext
        Loop
        objRS.Close
        set objRS = Nothing
        objConn.close
        set objConn = Nothing
%>
</select>
<hr>
Login Name: &nbsp &nbsp &nbsp <input type="text" size=8 maxsize=8
name="LOGIN"><br>
Password: &nbsp &nbsp &nbsp <input type="password" size=8 maxsize=8
name="PASSWORD"><p>
<input type="submit" value="Log-In"></form><form action="studentenroll.asp"
method=post>
</html>
```

## Appendix 5 – Professor's Toolbar Source Code

```
<!-- #include virtual = "BegAsp/adovbs.inc" -->
<html>
<title>Professor Toolbar</title>
<base target="main">
<BODY LINK="#FFFFFF" VLINK="#FFFFFF" ALINK="#FFFFFF">
<IMG SRC="inst_bar2.gif">
<CENTER>

<%

        Dim strUserID
        Dim strUserType
        Dim strPassword
        Dim strClassID
        Dim strSQL
        Dim strSQL2
        Dim objConn
        Dim objRS
        Dim objRS2

        strUserID = Session("USERID")
        strUserType = Session("USERTYPE")
        strPassword = Session("PASSWORD")
        strClassID = Session("CLASSID")

        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        strSQL                  = "SELECT * FROM ClassIndex "
        strSQL = strSQL & "WHERE TeacherID = '" & strUserID & "'"

        Set objRS = Server.CreateObject("ADODB.RecordSet")
        objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly, adCmdText

        If objRS.EOF Then
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%
```

```
        ElseIf objRS("Password") = strPassword Then

                objRS.Close

                Set objRS2 = Server.CreateObject("ADODB.RecordSet")

                strSQL                  = "SELECT * FROM ClassIndex "
                strSQL = strSQL & "WHERE ClassID = '" & strClassID & "';"

                strSQL2 = "SELECT * FROM Toolbar;"

                objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly,
adCmdText
                objRS2.Open strSQL2, objConn, adOpenStatic, adLockReadOnly,
adCmdText

                While Not objRS2.EOF
                        strTempString = objRS2("ToolID")

                        Response.Write "<a href="
                        %>"<%
                        Response.Write objRS2("Program")
                        %>"<%
                        Response.Write "> <img src="
                        %>"<%
                        Response.Write objRS2("ToolName") & ".gif"
                        %>"<%
                        Response.Write "> </a><p>"

                        objRS2.MoveNext

                Wend
                objRS.Close
                objRS2.Close

                Response.Write "</html>"

        Else
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%
        End If
%>
```

# Appendix 6 – Student's Toolbar Source Code

```
<!-- #include virtual = "BegAsp/adovbs.inc" -->
<html>
<title>Student Toolbar</title>
<base target="main">
<BODY LINK="#FFFFFF" VLINK="#FFFFFF" ALINK="#FFFFFF">
<IMG SRC="stu_bar2.gif">
<CENTER>
<%
        Dim strUserID
        Dim strUserType
        Dim strPassword
        Dim strClassID
        Dim strSQL
        Dim strSQL2
        Dim objConn
        Dim objRS
        Dim objRS2
        strUserID = Session("USERID")
        strUserType = Session("USERTYPE")
        strPassword = Session("PASSWORD")
        strClassID = Session("CLASSID")

        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        strSQL                  = "SELECT * FROM " & strClassID & "Members "
        strSQL = strSQL & "WHERE Login = '" & strUserID & "'"

        Set objRS = Server.CreateObject("ADODB.RecordSet")
        objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly, adCmdText

        If objRS.EOF Then
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%

        ElseIf objRS("Password") = strPassword Then

                objRS.Close
                Set objRS2 = Server.CreateObject("ADODB.RecordSet")

                strSQL                    = "SELECT * FROM ClassIndex "
                strSQL = strSQL & "WHERE ClassID = '" & strClassID & "';"

                strSQL2 = "SELECT * FROM Toolbar;"

                objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly,
        adCmdText
                objRS2.Open strSQL2, objConn, adOpenStatic, adLockReadOnly,
        adCmdText

                While Not objRS2.EOF
                        strTempString = objRS2("ToolID")

                        If (objRS(strTempString) = True) And (objRS2("UserType") =
        "Student") Then
                                Response.Write "<a href="
                                %>"<%
                                Response.Write objRS2("Program")
                                %>"<%
                                Response.Write "> <img src="
                                %>"<%
                                Response.Write objRS2("ToolName") & ".gif"
                                %>"<%
                                Response.Write "> </a><p>"
                        End If

                        objRS2.MoveNext

                Wend
                objRS.Close
                objRS2.Close

                Response.Write "</html>"

        Else
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%
        End If
%>
```

## Appendix 7 – Grade Display Source Code

```
<!-- #include virtual = "BegAsp/adovbs.inc" -->
<html>
<title>Grades</title>
<center>

<%

        Dim strUserID
        Dim strUserType
        Dim strPassword
        Dim strClassID
        Dim strSQL
        Dim strSQL2
        Dim strTempString
        Dim objConn
        Dim objRS
        Dim objRS2

        strUserID = Session("USERID")
        strUserType = Session("USERTYPE")
        strPassword = Session("PASSWORD")
        strClassID = Session("CLASSID")

        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        strSQL                  = "SELECT * FROM " & strClassID & "Members "
        strSQL = strSQL & "WHERE Login = '" & strUserID & "'"

        Set objRS = Server.CreateObject("ADODB.RecordSet")
        objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly, adCmdText

        If objRS.EOF Then
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%

        ElseIf objRS("Password") = strPassword Then

                Set objRS2 = Server.CreateObject("ADODB.RecordSet")
```

```
                    strSQL                  = "SELECT * FROM " & strClassID & "Grades "

                    objRS2.Open strSQL, objConn, adOpenStatic, adLockReadOnly,
adCmdText

                    Response.Write objRS("Login") & "'s Grades<p><hr><p>"
                    Response.Write "<table border=1 cols=4 width=80% >"
                    Response.Write "<tr><td></td><td>Points Received</td><td>Total
Points</td><td>Percentage</td></tr>"

                    While Not objRS2.EOF
                              Response.Write "<tr><td>" & objRS2("GradeID") & "</td>"
                              Response.Write "<td>" & objRS2("TotalScore") & "</td>"
                              strTempString = objRS2("GradeID")
                              Response.Write "<td>" & objRS(strTempString) & "</td>"
                              Response.Write "<td>" &
objRS(strTempString)/objRS2("TotalScore") * 100 & "</td></tr>"
                              objRS2.MoveNext
                    Wend

                    objRS.Close
                    objRS2.Close

                    Response.Write "</table></center></html>"

          Else
                    objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%

          End If

%>
```

## Appendix 8 – Class List Source Code

```
<!-- #include virtual = "BegAsp/adovbs.inc" -->
<html>
<title>Class Info</title>
<center>

<%

        Dim strUserID
        Dim strUserType
        Dim strPassword
        Dim strClassID
        Dim strSQL
        Dim strSQL2
        Dim strTempString
        Dim objConn
        Dim objRS
        Dim objRS2

        strUserID = Session("USERID")
        strPassword = Session("PASSWORD")
        strClassID = Session("CLASSID")

        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        strSQL                  = "SELECT * FROM " & strClassID & "Members "
        strSQL = strSQL & "WHERE Login = '" & strUserID & "'"

        Set objRS = Server.CreateObject("ADODB.RecordSet")
        objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly, adCmdText

        If objRS.EOF Then
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%

        ElseIf objRS("Password") = strPassword Then

                objRS.Close
                strSQL = "SELECT * FROM " & strClassID & "Members "
                objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly,
adCmdText

                Response.Write strClassID & " Class Info<p><hr><p>"
                Response.Write "<table border=1 cols=4 width=80% >"
                Response.Write "<b>"
                Response.Write "<tr><td>Login Name</td><td>Name of Student</td>"
                Response.Write "<td>E-Mail</td>"
                Response.Write "</tr></b>"

                While Not objRS.EOF
                        Response.Write "<tr><td>" & objRS("Login") & "</td>"
                        Response.Write "<td>" & objRS("Name") & "</td>"
                        Response.Write "<td>" & objRS("Email") & "</td>"
                        Response.Write "</tr>"
                        objRS.MoveNext
                Wend

                objRS.Close

                Response.Write "</table></center></html>"

        Else
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%
        End If
%>
```

# Appendix 9 – Password Source Code

```
<!-- #include virtual = "BegAsp/adovbs.inc" -->
<html>
<title>Change Password</title>
<center>

<%

        Dim strUserID
        Dim strUserType
        Dim strPassword
        Dim strClassID
        Dim strSQL
        Dim objConn
        Dim objRS
        Dim OldPass
        Dim PassChange
        Dim PassChange2

        strUserID = Session("USERID")
        strUserType = Session("USERTYPE")
        strPassword = Session("PASSWORD")
        strClassID = Session("CLASSID")
        strOldPass = Request.Form("OLDPASS")
        strPassChange = Request.Form("PASSCHANGE")
        strPassChange2 = Request.Form("PASSCHECK")

        Set objConn = Server.CreateObject("ADODB.Connection")
        objConn.Open "ClassInfo"

        strSQL                  = "SELECT * FROM ClassIndex "
        strSQL = strSQL & "WHERE TeacherID = '" & strUserID & "';"

        Set objRS = Server.CreateObject("ADODB.RecordSet")
        objRS.Open strSQL, objConn, adOpenStatic, adLockReadOnly, adCmdText

        If objRS.EOF Then
                objRS.Close

%>
Error:  Must start from Login Screen
</html>
<%
```

```
        ElseIf objRS("Password") = strPassword AND strUserType = "Professor" Then

                If Not strOldPass = strPassword Then

%>
<html>
You entered the wrong original password!
</html>
<%

                ElseIf strPassChange = strPassChange2 Then

                        strSQL = "UPDATE ClassIndex SET "
                        strSQL = strSQL & "ClassIndex.Password = '" & strPassChange &
"' "
                        strSQL = strSQL & "WHERE TeacherID = '" & strUserID & "';"

                        objConn.Execute strSQL

                        Session("PASSWORD") = strPassChange

%>
Your Password has been changed.
</html>
<%
                Else
%>
You must put the same password for both fields!
</html>
<%
                End If

        objRS.Close
        Set objRS = Nothing
        objConn.Close
        Set objConn = Nothing

        Else
%>
Error:  Must start from Login Screen
</html>
<%

End If
%>
```

# Appendix 10 – Chat Server Source Code

```
#!/Perl/bin/perl

#>--------------------------------------<#
#| CONFIGURATION (MODIFY THIS SECTION)  |#
#>--------------------------------------<#

# place the absolute path to your chat room message files
# here (include a trailing slash).

        $filepath='C:/Inetpub/wwwroot/cgi-bin/messages/';

# place the file extention of your chat room message files
# here, including the dot.  It is probably .htm or .html

        $filext='.html';

# setting this flag will make messages scroll from top to
# bottom, making the script compatible with EVERY browser.
# See the readme.txt file under OPTIONS for more details!

        $iecompatible=0;

#>--------------------------------------<#
#| Main Program                         |#
#| (all it does is call subprograms)    |#
#>--------------------------------------<#

print "Content-type: text/html\nPragma: no-cache\n\n";
print "<html><title></title><BODY BGCOLOR=#000080 TEXT=#FFFFFF>\n";
&getform;
if (&getoldfile) {
        &gettime;
        &printform;
        if ($form{'message'} ne "") { &printnewfile; }
        else { &printnoframes; }
}
else
{
        print "The room you entered <I>$form{'room'}</I> does not exist.<BR>\n";
        print "Tell the server administrator to check the file:
<I>$filepath$form{'room'}$filext</I>\n";
}
print "</font></body></html>\n";
```

```
exit(0);

#>---------------------------------------<#
#| Sub getform - reads form data         |#
#>---------------------------------------<#

sub getform {
        $buffer = "";
        read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
        @pairs=split(/&/,$buffer);
        foreach $pair (@pairs)
        {
                @a = split(/=/,$pair);
                $name=$a[0];
                $value=$a[1];
                $value =~ s/\+/ /g;
                $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
                $value =~ s/~!/ ~!/g;
                $value =~ s/\</\&lt\;/g;  # html tag removal (remove these lines to enable
HTML tags in messages)
                $value =~ s/\>/\&gt\;/g;  # html tag removal (remove these lines to enable
HTML tags in messages)
                $value =~ s/[\r\n]//g;
                push (@data,$name);
                push (@data,$value);
        }
        %form=@data;
        %form;
}

#>---------------------------------------<#
#| Sub getoldfile - reads old HTML        |#
#| messages file and returns 0 if not    |#
#| found or 1 if found                   |#
#>---------------------------------------<#

sub getoldfile {
        $form{'room'} =~ s/\W//g;
        return 0 unless open(HTMLOLD, "$filepath$form{'room'}$filext");
        @lines=<HTMLOLD>;
        close(HTMLOLD);
        return 1;
}

#>---------------------------------------<#
#| Sub gettime - reads system time       |#
```

```
#>--------------------------------------<#

sub gettime {
	$now_string = localtime;
	@thetime = split(/ +/,$now_string);
	@theclock = split(/:/,$thetime[3]);
	$ampm = 'am';
	if ($theclock[0] > 11)
	{ $ampm = 'pm'; }
	if ($theclock[0] == 0)
	{ $theclock[0] = 12; }
	if ($theclock[0] > 12)
	{ $theclock[0] -= 12; }
	else
	{ $theclock[0] += 0; }
}

#>--------------------------------------<#
#| Sub printform - prints new form       |#
#>--------------------------------------<#

sub printform {
	if ($form{'logoff'} eq '1')
	{
		print << "EOF";
		<CENTER></CENTER><BR><HR><FONT SIZE=-1>
EOF
	}
	else
	{
		print << "EOF";
		<CENTER><TABLE CELLSPACING=0 CELLPADDING=0>
		<TR><TD>
		<nobr><FORM ACTION="$ENV{'SCRIPT_NAME'}"
METHOD="POST">
		Your message: <input name=username type=hidden
value="$form{'username'}">
		<input name=room type=hidden value="$form{'room'}">
		<input type=text name=message size=35>
		<input type=submit value="Post This">
		</form></nobr>
		</TD><TD>
		<nobr><FORM ACTION="$ENV{'SCRIPT_NAME'}"
METHOD="POST">
		<input name=username type=hidden value="$form{'username'}">
		<input name=room type=hidden value="$form{'room'}">
```

```
                <input name=logoff type=hidden value=1>
                <input type=hidden name=message value="I logged off!">
                <input type=submit value="Logoff">
                </form></nobr>
                </TD></TR>
                </TABLE></CENTER><BR><HR>
                <FONT SIZE=-2>Hit "post" without entering a message to refresh the
screen...</FONT><FONT SIZE=-1>
EOF
        }
}

#>--------------------------------------<#
#| Sub printnewfile - prints new HTML    |#
#| messages file                         |#
#>--------------------------------------<#

sub printnewfile {
        $newmessage = "<P><B>$form{'username'}</B> says,\"$form{'message'}\"
($thetime[0] $theclock[0]:$theclock[1]$ampm)\n";
        open (NEW, ">$filepath$form{'room'}$filext");
        print NEW "<HTML><HEAD><META HTTP-EQUIV=Refresh
CONTENT=4></HEAD><BODY BGCOLOR=#FFFFFF>\n";
        if ($iecompatible) {
                print NEW $newmessage;
                print $newmessage;
                for ($i = 1; $i < 15; $i++)
                {
                        print NEW "$lines[$i]";
                        print "$lines[$i]";
                }
                print NEW "<BR><FONT COLOR=#FFFFFF></FONT></BODY>\n";
        }
        else {
                for ($i = 2; $i < 16; $i++)
                {
                        print NEW "$lines[$i]";
                        print "$lines[$i]";
                }
                print NEW $newmessage;
                print $newmessage;
                print NEW "<BR><FONT
COLOR=#FFFFFF></A></FONT></BODY>\n";
        }
        close NEW;
}
```

```
#>---------------------------------------<#
#| Sub printnoframes - refreshes screen  |#
#| if no messages are posted             |#
#>---------------------------------------<#

sub printnoframes {
        for ($i = 1; $i < 16; $i++)
        {
                print "$lines[$i]";
        }
}
```

## Appendix 11 – Quiz Source Code

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="GENERATOR" content="Mozilla/4.5 [en] (WinNT; I) [Netscape]">
  <meta name="Author" content="Yung Giang">
  <title>ES2001 Exam#1</title>
<script>
<!-- hide
var ans = new Array;
var done = new Array;
var right = 0;
var wrong = 0;
var per = 0;
var percent = 0;
var temp = 0;

// Enter answers here!
ans[1] = "b";
ans[2] = "b";
ans[3] = "b";
ans[4] = "b";
ans[5] = "a";
ans[6] = "a";
ans[7] = "d";
ans[8] = "b";
ans[9] = "c";
ans[10] = "b";
ans[11] = "c";
ans[12] = "b";
ans[13] = "a";
ans[14] = "c";
ans[15] = "c";
ans[16] = "a";
ans[17] = "b";
ans[18] = "c";
ans[19] = "a";
ans[20] = "b";

function Engine(question, answer) {

        if (answer != ans[question]) {
                if (!done[question]) {
                        done[question] = -1;
                        wrong+=1;
                        per=(right/(right+wrong)*100);
                        percent=Math.round(per);
                        /*

        win2=open("","window2",'width=250,height=150','scrollbars=1','sizeable=no')
                        win2.document.open
                        win2.document.writeln("<b>Incorrect")
                        win2.document.writeln("<br>")
                        win2.document.writeln("<br>Correct Answer= " + ans[question])
                        win2.document.writeln("<br>")
                        win2.document.writeln("<br>\n\nRight: " + right)
                        win2.document.writeln("\n\nWrong: " + wrong)
                        win2.document.writeln("\n\nScore: " + percent)
                        win2.document.writeln("<br>")
                        win2.document.writeln("<br><u>Close This Window
Now!</u></b>");
                        win2.document.close()
                        */
                        }
                else {

        win2=open("","window2",'width=250,height=150','scrollbars=1','sizeable=no')
                        win2.document.open
                        win2.document.writeln("<b>Already Answered This......");
                        /*
                        win2.document.writeln("<br>")
                        win2.document.writeln("<br>Correct Answer Was= " +
ans[question])
                        win2.document.writeln("<br>")
                        win2.document.writeln("<br><u>Close Window Now!</u></b>");
                        */
                        win2.document.close()
                        }
                }
        else {
                if (!done[question]) {
                        done[question] = -1;
                        right+=1;
                        per=(right/(right+wrong)*100);
                        percent=Math.round(per);
                        /*

        win2=open("","window2",'width=250,height=150','scrollbars=1','sizeable=no')
```

```
                    win2.document.open
                    win2.document.writeln("<b>Correct")
                    win2.document.writeln("<br>")
                    win2.document.writeln("<br>\n\nRight: " + right)
                    win2.document.writeln("\n\nWrong: " + wrong)
                    win2.document.writeln("\n\nScore: " + percent)
                    win2.document.writeln("<br>")
                    win2.document.writeln("<br><u>Close Window Now!</u></b>");
                    win2.document.close()
                    */
                    }
                else {

        win2=open("","window2",'width=250,height=150','scrollbars=1','sizeable=no')
                    win2.document.open
                    win2.document.writeln("<b>Already Answered This......");
                    /*
                    win2.document.writeln("<br>")
                    win2.document.writeln("<br>Correct Answer Was= " +
ans[question])
                    win2.document.writeln("<br>")
                    win2.document.writeln("<br><u>Close Window Now!</u></b>");
                    */
                    win2.document.close()
                    }
                }
}
function button(click){
        if (click == 1){
        win2=open("","window2",'width=250,height=430','scrollbars=1','sizeable=yes')
        win2.document.open
        win2.document.writeln("\n\nRight: " + right)
        win2.document.writeln("\n\nWrong: " + wrong)
        win2.document.writeln("\n\nScore: " + percent)
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 1: " + ans[1])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 2: " + ans[2])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 3: " + ans[3])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 4: " + ans[4])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 5: " + ans[5])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 6: " + ans[6])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 7: " + ans[7])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 8: " + ans[8])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 9: " + ans[9])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 10: " + ans[10])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 11: " + ans[11])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 12: " + ans[12])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 13: " + ans[13])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 14: " + ans[14])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 15: " + ans[15])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 16: " + ans[16])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 17: " + ans[17])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 18: " + ans[18])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 19: " + ans[19])
        win2.document.writeln("<br>")
        win2.document.writeln("\n\nAnswer 20: " + ans[20])
        win2.document.writeln("<br>")
        win2.document.writeln();
        win2.document.close()
        }
/*
        if (click == 1){
        win2=open("","window2",'width=250,height=150','scrollbars=1','sizeable=no')
        win2.document.open
        while (temp != question) {
                win2.document.writeln("<br>" + temp "= " + ans[temp])
                temp = temp+1;
                }
        win2.document.close()
        }
*/
}

//-->
```

```
</script>
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#0000EE" vlink="#551A8B"
alink="#FF0000" background="../pageimg/back1.gif" nosave>
<a NAME="top"></a><script language="javascript">
// Netscape 3.0 compatibility test (for javascript image swapping)
compat = false;
if( parseInt( navigator.appVersion ) >= 3 ) { compat = true; }
// cache images for quick swapping
if( compat )
{
HOMEoff = new Image;
HOMEoff.src = "../pageimg/home1.gif";
HOMEon = new Image;
HOMEon.src = "../pageimg/home2.gif";
INDEXoff = new Image;
INDEXoff.src = "../pageimg/index1.gif";
INDEXon = new Image;
INDEXon.src = "../pageimg/index2.gif";
TOPoff = new Image;
TOPoff.src = "../pageimg/top1.gif";
TOPon = new Image;
TOPon.src = "../pageimg/top2.gif";

}

// swap images using the cached images

function glow(x, y)
{
  if( compat ) { document.images[x].src=eval(y+'.src'); }
}
</script>

<table BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH="650" >
<tr>
<td><img SRC="../pageimg/invisible.gif" NOSAVE height=1 width=55></td>
<td></td>

<! Enter Or Change Header here>
<td>
<center><form><b><font color="#FF0000">ES2001 Exam#1:::Shivkumar
</font></b>
<p><font color="#000000">The score is purely for your benefit and is not
recorded.&nbsp;</font>
```

```
<br><font color="#000000">To start the exam:&nbsp;&nbsp; </font><font
color="#FF0000">hit
reload and then the button below<b>:</b></font>
<br><font color="#FF0000">Make sure you close the grading window before
attempting the next question.</font>
<br><input TYPE="RESET", Value='Clear All Check Marks'></center>
</td>
</tr>

<tr>
<td></td><td COLSPAN="2">&nbsp;
<center><br></center></td></tr>

<tr>
<td></td>
<td></td>
<td>
<center><noscript>JavaScript is <b><i>disabled</i></b>. Get Netscape 3.0
or turn it on!</noscript></center>
</td>
</tr>

<!Enter  or Change Questions here>
<!Question one>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">1.</font></b></td>
<td>The electronic of Mg is:
<p>
<input type=checkbox value="a" onClick="Engine(1,
this.value)">1s<sup>2</sup>2s<sup>2</sup>2p<sup>6</sup>3s
<br>
<input type=checkbox value="b" onClick="Engine(1,
this.value)">1s<sup>2</sup>2s<sup>2</sup>2p<sup>6</sup>3s<sup>2</sup>
<br>
<input type=checkbox value="c" onClick="Engine(1,
this.value)">1s<sup>2</sup>2s<sup>2</sup>2p<sup>8</sup>
<br>
<input type=checkbox value="d" onClick="Engine(1, this.value)">None of the Above
</td>
</tr>

<!question two>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">2.</font></b></td>
```

```
<td>The valence for Mg is:
<p>
<input type=checkbox value="a" onClick="Engine(2, this.value)">1
<br>
<input type=checkbox value="b" onClick="Engine(2, this.value)">2
<br>
<input type=checkbox value="c" onClick="Engine(2, this.value)">3
<br>
<input type=checkbox value="d" onClick="Engine(2, this.value)">None of the Above
</td>
</tr>

<!Question number 3 >
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">3.</font></b></td>
<td>Ionic bond can exist in:
<p>
<input type=checkbox value="a" onClick="Engine(3, this.value)">Al
<br>
<input type=checkbox value="b" onClick="Engine(3, this.value)">CaO
<br>
<input type=checkbox value="c" onClick="Engine(3, this.value)">Nylon
<br>
<input type=checkbox value="d" onClick="Engine(3, this.value)">None of the Above
</td>
</tr>

<!Question number 4>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">4.</font></b></td>
<td>Materials with ionic bonding have:
<p>
<input type=checkbox value="a" onClick="Engine(4, this.value)">High density
<br>
<input type=checkbox value="b" onClick="Engine(4, this.value)">High strength
<br>
<input type=checkbox value="c" onClick="Engine(4, this.value)">High melting point
<br>
<input type=checkbox value="d" onClick="Engine(4, this.value)">None of the Above
</td>
</tr>

<!Question number 5>
<tr>
```

```
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">5.</font></b></td>
<td>Covalent bonds may be present in:
<p>
<input type=checkbox value="a" onClick="Engine(5, this.value)">Benzene
<br>
<input type=checkbox value="b" onClick="Engine(5, this.value)">Mg
<br>
<input type=checkbox value="c" onClick="Engine(5, this.value)">Si
<br>
<input type=checkbox value="d" onClick="Engine(5, this.value)">None of the Above
</td>
</tr>

<!Question number 6>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">6.</font></b></td>
<td>Materials with predominantly covalent bonds are:
<p>
<input type=checkbox value="a" onClick="Engine(6, this.value)">Very strong
<br>
<input type=checkbox value="b" onClick="Engine(6, this.value)">Brittle
<br>
<input type=checkbox value="c" onClick="Engine(6, this.value)">Poor thermal
conductors
<br>
<input type=checkbox value="d" onClick="Engine(6, this.value)">None of the Above
</td>
</tr>

<!Question number 7>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">7.</font></b></td>
<td>Polymers have low strength and melting point compared to metals because:
<p>
<input type=checkbox value="a" onClick="Engine(7, this.value)">They contain many
defects
<br>
<input type=checkbox value="b" onClick="Engine(7, this.value)">The basic element in
the structure is Carbon
<br>
<input type=checkbox value="c" onClick="Engine(7, this.value)">They are light
<br>
```

```
<input type=checkbox value="d" onClick="Engine(7, this.value)">The molecules are
held together by weak secondary bonds
</td>
</tr>

<!Question number 8>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">8.</font></b></td>
<td>The close packed plane for a BCC structure is:
<p>
<input type=checkbox value="a" onClick="Engine(8, this.value)">(100)
<br>
<input type=checkbox value="b" onClick="Engine(8, this.value)">(110)
<br>
<input type=checkbox value="c" onClick="Engine(8, this.value)">(111)
<br>
<input type=checkbox value="d" onClick="Engine(8, this.value)">None of the Above
</td>
</tr>

<!Question number 9>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">9.</font></b></td>
<td>The close packed direction for a BCC structure is:
<p>
<input type=checkbox value="a" onClick="Engine(9, this.value)">[100]
<br>
<input type=checkbox value="b" onClick="Engine(9, this.value)">[110]
<br>
<input type=checkbox value="c" onClick="Engine(9, this.value)">[111]
<br>
<input type=checkbox value="d" onClick="Engine(9, this.value)">None of the Above
</td>
</tr>

<!Question number 10>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">10.</font></b></td>
<td>The close packed direction for a FCC structure is:
<p>
<input type=checkbox value="a" onClick="Engine(10, this.value)">[100]
<br>
<input type=checkbox value="b" onClick="Engine(10, this.value)">[110]
```

```
<br>
<input type=checkbox value="c" onClick="Engine(10, this.value)">[111]
<br>
<input type=checkbox value="d" onClick="Engine(10, this.value)">None of the Above
</td>
</tr>

<!Question number 11>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">11.</font></b></td>
<td>The close packed plane for a FCC structure is:
<p>
<input type=checkbox value="a" onClick="Engine(11, this.value)">(100)
<br>
<input type=checkbox value="b" onClick="Engine(11, this.value)">(110)
<br>
<input type=checkbox value="c" onClick="Engine(11, this.value)">(111)
<br>
<input type=checkbox value="d" onClick="Engine(11, this.value)">None of the Above
</td>
</tr>

<!Question number 12>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">12.</font></b></td>
<td>the atomic packing factor for an FCC structure is:
<p>
<input type=checkbox value="a" onClick="Engine(12, this.value)">greater than 1
<br>
<input type=checkbox value="b" onClick="Engine(12, this.value)">is equal to that for a
HCP structure
<br>
<input type=checkbox value="c" onClick="Engine(12, this.value)">is greater than the
value for BCC structure
<br>
<input type=checkbox value="d" onClick="Engine(12, this.value)">is equal to 0.2
</td>
</tr>

<!Question number 13>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">13.</font></b></td>
<td>MgO has a melting point of 2800C because:
```

```
<p>
<input type=checkbox value="a" onClick="Engine(13, this.value)">it is held together by
ionic bonds
<br>
<input type=checkbox value="b" onClick="Engine(13, this.value)">it is held together by
covalent bond
<br>
<input type=checkbox value="c" onClick="Engine(13, this.value)">it is held together by
van der Walls bonds
<br>
<input type=checkbox value="d" onClick="Engine(13, this.value)">None of the Above
</td>
</tr>

<!Question number 14>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">14.</font></b></td>
<td>Potassium has a lower melting point than Co because:
<p>
<input type=checkbox value="a" onClick="Engine(14, this.value)">its atoms are small
<br>
<input type=checkbox value="b" onClick="Engine(14, this.value)">it has a lower atomic
weight
<br>
<input type=checkbox value="c" onClick="Engine(14, this.value)">it has a lower
number f electrons available to participate in metallic bonding
<br>
<input type=checkbox value="d" onClick="Engine(14, this.value)">None of the Above
</td>
</tr>

<!Question number 15>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">15.</font></b></td>
<td>The valence of metallic elements is generally:
<p>
<input type=checkbox value="a" onClick="Engine(15, this.value)">greater than 4
<br>
<input type=checkbox value="b" onClick="Engine(15, this.value)">equal to 4
<br>
<input type=checkbox value="c" onClick="Engine(15, this.value)">less than or equal to
3
<br>
<input type=checkbox value="d" onClick="Engine(15, this.value)">very high
```

```
</td>
</tr>

<!Question number 16>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">16.</font></b></td>
<td>Nonmetallic elements:
<p>
<input type=checkbox value="a" onClick="Engine(16, this.value)">have a valence
greater than 4
<br>
<input type=checkbox value="b" onClick="Engine(16, this.value)">can accept electrons
to form anios
<br>
<input type=checkbox value="c" onClick="Engine(16, this.value)">have less than 3
valence electrons
<br>
<input type=checkbox value="d" onClick="Engine(16, this.value)">have high melting
points and strengths
</td>
</tr>

<!Question number 17>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">17.</font></b></td>
<td>Heating water above the boiling point leads to steam formation. This corresponds to
the:
<p>
<input type=checkbox value="a" onClick="Engine(17, this.value)">breaking of water
molecule into hydrogen and oxygen
<br>
<input type=checkbox value="b" onClick="Engine(17, this.value)">breaking of the weak
van der Walls bonds that exist between water molecules
<br>
<input type=checkbox value="c" onClick="Engine(17, this.value)">breaking of the
covalent bonds between hydrogen and oxygen
<br>
<input type=checkbox value="d" onClick="Engine(17, this.value)">None of the Above
</td>
</tr>

<!Question number 18>
<tr>
<td></td>
```

```
<td VALIGN=TOP><b><font color="#0000FF">18.</font></b></td>
<td>A polymorphic or allotropic transformation occurs upon heating iron to 912C. This
transformation leads to:
<p>
<input type=checkbox value="a" onClick="Engine(18, this.value)">melting of iron
<br>
<input type=checkbox value="b" onClick="Engine(18, this.value)">a change in color
<br>
<input type=checkbox value="c" onClick="Engine(18, this.value)">a change in crystal
structure from BCC to FCC
<br>
<input type=checkbox value="d" onClick="Engine(18, this.value)">Very high
</td>
</tr>

<!Question number 19>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">19.</font></b></td>
<td>The structure of polymers or plastics can be viewed as:
<p>
<input type=checkbox value="a" onClick="Engine(19, this.value)">covalently bonded
long molecules held together by van der Walls bonds
<br>
<input type=checkbox value="b" onClick="Engine(19, this.value)">ionically bonded
mers
<br>
<input type=checkbox value="c" onClick="Engine(19, this.value)">covalently bonded
mers
<br>
<input type=checkbox value="d" onClick="Engine(19, this.value)">permanently
polarized molecules
</td>
</tr>

<!Question number 20>
<tr>
<td></td>
<td VALIGN=TOP><b><font color="#0000FF">20.</font></b></td>
<td>Materials with covalent bonds are generally good insulators because:
<p>
<input type=checkbox value="a" onClick="Engine(20, this.value)">they have high
strength
<br>
<input type=checkbox value="b" onClick="Engine(20, this.value)">for electrons to
move, the covalent bonds must be broken and this requires high voltages or temperatures
```

```
<br>
<input type=checkbox value="c" onClick="Engine(20, this.value)">they have too few
valence electrons
<br>
<input type=checkbox value="d" onClick="Engine(20, this.value)">they have 3 electrons
in the 3s orbital.
<br>
</td>
</tr>
</table>
<br>
<center><input type=button value="Result and Answers" onclick="button(1)"></center>
</body>
</html>
```