A Programming Laboratory for Electronic Commerce

Alfred C. Weaver
Department of Computer Science, University of Virginia,
151 Engineer's Way, Charlottesville, VA 22904 weaver@virginia.edu

Abstract – With support from NSF DUE CCLI [1], we developed a programming laboratory to provide interactive experience with the tools, techniques, and principles of e-commerce. The first lab explored the wide range of web innovation, the last lab was a mock patent infringement trial, and the eleven other labs were chosen from a set of independent modules on HTML, CSS, JavaScript, Perl, CGI, regular expressions, SQL, ASP, PHP, Flash, XML, and C#. The first lab offering used weekly, three-hour physical labs; students enjoyed the material but rejected the fixed time weekly format. The next offering will transition to the course to utilize virtual labs that will better align the programming experience with the 24/7 nature of the Internet and e-commerce.

Index Terms – electronic commerce, software laboratory, virtual labs

Motivation

CS453 Electronic Commerce Technologies [2] is an elective course for third- and fourth-year Computer Science majors at the University of Virginia (CS is in the School of Engineering and Applied Science). It began as a lecture-only course in 1995 when e-commerce was new, and has evolved each year such that now we cover both technology and some business strategy. The typical student is already an accomplished C++ programmer.

As the course evolved, the traditional technique of bi-weekly programming homeworks seemed increasingly inadequate as a way of preparing students for creating an e-store with sufficient functionality and robustness that it could be trusted to conduct business and manage secure financial transactions over the public Internet. To increase the realism of the e-commerce experience, I proposed and the National Science Foundation funded a DUE CCLI proof-of-concept project to create a physical, three-hour, weekly programming laboratory, staffed by the instructor and teaching assistants, to provide this missing hands-on experience.

While the students viewed the lab content positively, they objected to the weekly Wednesday night labs (this aspect is discussed in more detail later). Thus we conceived, and the University of Virginia is currently funding, a second project to make the labs virtual rather than physical. Virtual labs require more detailed instructions because the expectation is that students will be using them when there is no staff member available for questions. This naturally leads to questions about how to provide staff support for virtual labs, and this too is discussed later.

This paper reports our achievements, errors, and recommendations for those who might wish to follow a similar strategy, whether for e-commerce or some other technical topic amenable to web-based exposition. For e-commerce in particular, the instructional modules that we developed are publicly available and we invite other instructors to use, improve, and return materials to our repository. The submission of new modules and new topics is enthusiastically encouraged.

Goals

Our goals for the original hands-on laboratory were:
(1) Select technical topics that can be introduced, examined, and explained in lecture but whose mastery will occur in the programming laboratory;
(2) Create laboratory modules for selected topics, such that students will have a "guided" experience via exercises of increasing difficulty and sophistication;
(3) Develop all material in a modular fashion such that units can be easily imported and exported from a master web-based repository, with the expectation that over time the collection will grow to the point that a future instructor can customize his or her own course by picking and choosing from many available lab modules on many available topics (this goal anticipates the participation of many additional faculty to achieve the necessary volume of lab material).

Now that we have embarked on a “virtual lab” version of the course, a future goal is to compare the educational outcomes of the virtual labs vs. physical labs.

Core Topics

The core topics for both lecture and lab were selected by our stakeholders, who include the CS faculty at U.Va., CS faculty volunteers from Longwood University, over one hundred students in previous versions of CS453, and our lab development team consisting of myself, one graduate and three undergraduate students. While there was spirited discussion regarding which topics to choose, it was clear to all concerned that this course was a work in progress, that any selection of material made by one person was likely to be non-optimal from the viewpoint of a different person, and that the real value of the concept lay not in teaching a specific set of topics and techniques, but rather in building a repository of
useful material for sufficiently many topics that future instructors could pick and choose to create their own customized course.

The lecture topics most recently used are shown in Table I. These topics get reordered or replaced each semester as the instructor’s and students’ interests change and as technology progresses.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>History of Internet and WWW</td>
</tr>
<tr>
<td>2</td>
<td>Web innovation (who, what, and when)</td>
</tr>
<tr>
<td>3</td>
<td>Case study (eBay)</td>
</tr>
<tr>
<td>4</td>
<td>HTML, CSS, and dynamic HTML</td>
</tr>
<tr>
<td>5</td>
<td>JavaScript, events; client-side control</td>
</tr>
<tr>
<td>6</td>
<td>Cryptography</td>
</tr>
<tr>
<td>7</td>
<td>SET, SSL, and PGP</td>
</tr>
<tr>
<td>8</td>
<td>Perl and CGI</td>
</tr>
<tr>
<td>9</td>
<td>Electronic payment systems</td>
</tr>
<tr>
<td>10</td>
<td>SQL for inventory databases</td>
</tr>
<tr>
<td>11</td>
<td>ASP (or PHP)</td>
</tr>
<tr>
<td>12</td>
<td>Starting an e-commerce company</td>
</tr>
<tr>
<td>13</td>
<td>Intellectual property (copyrights, patents)</td>
</tr>
<tr>
<td>14</td>
<td>Patent infringement</td>
</tr>
<tr>
<td>15</td>
<td>Reliability (viruses, worms, denial of service)</td>
</tr>
<tr>
<td>16</td>
<td>Ethics</td>
</tr>
<tr>
<td>17</td>
<td>Wireless Internet access (devices and standards)</td>
</tr>
<tr>
<td>18</td>
<td>Biometric authentication</td>
</tr>
<tr>
<td>19</td>
<td>Smartcards</td>
</tr>
<tr>
<td>20</td>
<td>Future of e-commerce</td>
</tr>
</tbody>
</table>

Although the focus of this paper is on the lab topics and programming exercises, let me set the stage for them by explaining the first and last laboratory experiences. Even when we switched from physical to virtual programming labs, these first and last labs were retained as required, three-hour, physical meetings.

**First Laboratory**

Our stakeholders’ strong recommendation was that the first lab should be experiential, thought provoking, and should emphasize a wide range of web innovation. We thought it crucial that students who were already very familiar with popular websites (e.g., 90% had already bought or sold something on eBay) be challenged to think not about what it did (which they already knew) but instead about how it accomplished some task that they took for granted. We identified several suitable experiences and then used as many of those as would fit in our first three-hour lab. An essential characteristic of each example was that it would illustrate a technique, technology, principle, or requirement that we would later revisit in class.

Here are three examples:

1. We divided our 50 students into teams of three or four and gave each team an object to sell or auction on eBay. After loading their item descriptions, pictures, payment terms, etc., we turned to discussion. Why does eBay force a seller to prove his/her identity before allowing that user to sell? How does eBay verify the proffered identity documents, and what information must be available to eBay in order for them to do that? How does eBay secure transmission of sensitive information such as credit card numbers? (Here a “view source” exposes SSL and sets the stage for future discussion of cryptography and electronic payment systems). How does eBay keep its vast collection of customer financial information from being stolen? (All such data are encrypted and kept on a server not connected to the Internet, which sets the stage for our future topics of trust and reliability.)

2. Download an MP3 and play it, then use RealPlayer to watch a movie trailer. Discussion points: What’s the difference between file downloading and streaming? What causes the jitter in streaming multimedia? Was your MP3 download legal? What is the impact of music file sharing on the artist, production company, distribution company, and society in general? Outline the rise and fall and rebirth of Napster. This discussion provides context for our future discussion about ethics, protection of intellectual property, and digital rights management.

3. Ask Google a straight-forward question such as “how do fingerprint scanners work?” How did Google assemble 23,200 responses? This foreshadows spiders and webcrawlers. How did Google decide which links were most relevant? This introduces Google’s patented relevancy ranking scheme. Ask Google a nonsense question such as “who is the current king of the united states?” Why were there 766,000 responses to a question whose answer is “nobody”? Discuss the limitations of text-based search that lacks semantic understanding. How would you search audio or video files if they were not tagged with a markup language?

**Last Laboratory**

Our stakeholders recommended that the last lab should be as participatory as the first. Recognizing that the protection of intellectual property (IP) is a growing concern in a digital society, we devote lecture time to the issues of IP generation and protection via the mechanisms of copyrights, patents, trademarks, and trade secrets. To make the concepts seem more real, we examine a few patents that disclose inventions likely to be known or used by the students. This allows us to discuss the crucial attributes of valid patents (e.g., novelty, usefulness, non-obviousness), the legal criteria for patent infringement, and how accused infringers attempt to avoid penalties by proving non-infringement or by attacking the fundamental validity of the patent, typically by asserting obviousness.
In preparation for our final laboratory exercise, a mock trial for patent infringement, we introduce U.S. Patent 5,960,411 to Hartman et al. [3] and assigned to Amazon.com, and colloquially referred to as the “Amazon one-click patent” because its claims encompass the placing of an order over the Internet using any of several “one-action” techniques such as one mouse click, one keypress, one sound, etc. We identify a commercial product that might infringe the “one-click” patent. Well in advance of our mock trial, selected students form prosecution and defense teams and study the patent, patent law, and the accused infringing device. The mock trial allows the prosecution to present its evidence of infringement, the defense presents its arguments or non-infringement and patent invalidity, and the remaining students serve as the jury.

Following the mock trial, we hear the true story of the Amazon vs. Barnes and Noble patent infringement case from a patent attorney who actually participated in that case. In the three years that we’ve held the mock trial as our final lab, students consistently rate it as their favorite lab.

**TECHNOLOGY LABORATORIES**

Our stakeholders recommended that the technology labs [4] initially cover HTML, CSS, JavaScript, Perl, CGI, SQL, ASP, and Flash; these were developed with NSF funding. Subsequent to that grant we have added PHP, C# and XML labs, and we are preparing for a future expansion into web services. See Figure 1 for the lab’s homepage.

![Figure 1: Programming Lab Homepage](image)

Each topic is a module unto itself, and all modules follow a common format by presenting material at three levels of difficulty: beginner, intermediate, and advanced. Within a topical module, each lab exercise presents some programming concept, provides a code snippet that illustrates the concept, and then proposes a specific coding problem that needs the identified concept for its solution.

Clicking on a topic, say JavaScript, reveals that material as shown in Figure 2: the list of individual exercises, labeled by level of difficulty (beginning, intermediate, advanced) and by topic (e.g., forms, checkboxes, cookies). Every topic includes an “answers” folder that provides the solutions to all problems and, if needed, an “extras” folder that provides useful files (e.g., bitmaps, databases) that the student can use without first having to create them.

![Figure 2: Exercises and Answers for JavaScript](image)

Clicking on any of the exercises listed, say the JavaScript exercise on handling radio buttons, displays an assignment as shown in Figure 3. Each exercise provides an overview of the principle or technique being demonstrated, code that illustrates what is going on and also provides a template or harness for code that the student will write, and then a specific assignment. The topical coverage of the labs developed thus far is shown in Table II.

**HOMEWORKS**

Each lab topic culminates in a homework that seeks to integrate the material presented piecemeal through the laboratory exercises. In Fall 2003 those homeworks were:

- **JavaScript** – a casino website that plays poker using Las Vegas rules.
- **Perl/CGI** – "five-minute auctions", a site that allows posting of a title and description of a good for auction; increasing bids are accepted for five minutes; auction closes and winner is notified after five minutes duration.
- **SQL** – develop a database for a store's inventory
- **ASP or PHP** – utilizing the SQL database, create an electronic store for computer supplies; support on-line ordering, shopping cart management, electronic catalog, inventory management, and billing.

**THE VIRTUAL LAB**

Students appreciate the 24/7 nature of the Internet, and student evaluation of the first offering of the physical lab showed conclusively that they wanted to make the lab experience virtual rather than physical. As one student opined, "My schedule is very hectic. The Wednesday night lab hours were inconvenient in view of assignments in other classes and..."
extracurricular activities. I want to use the lab resources on my schedule, not yours."

If the student learning resulting from a virtual lab approach is equivalent or better to that which results from physical labs, then the virtual labs would have another advantage (in addition to student preference) in that we could utilize computing resources during non-traditional time (say 10 pm-8 am when the physical labs are closed). To test the hypothesis that a virtual lab approach is equivalent, U.Va.’s Teaching + Technology program is funding a project to create a virtual lab version of the course. That rewrite is currently underway and is expected to be ready for use in Fall 2004.

The challenge of the virtual lab is not so much in the lab content as it is in the requisite ancillary resources. A student working a lab at 3 a.m. is not going to have reliable access to the course staff. In future course offerings we hope to evaluate the following to determine the best combination of support services for the virtual lab:

- classical, fixed-time office hours for the course staff
- electronic, asynchronous contact with the course staff via email
- audio/video consultation with the course staff via Microsoft NetMeeting
- rapidly updated (by the staff) FAQs for each lab topic
- free-form electronic interchanges (bulletin boards and email forums) among students to encourage them to help each other as they discover questions and answers.

WHAT DID WE LEARN?

(1) All students underwent an extensive exit survey to gather their comments, complaints, suggestions, and innovations. Without exception, every student evaluated the programming lab as a valuable learning experience.

(2) Some common themes were the instant availability of help from the TAs, the incremental nature of learning each new subject (beginner, intermediate, advanced), and the resulting ability to simply skip topics that a student already understood.

(3) Two-thirds of the students objected to the three-hour Wednesday night lab time. These students recommended presenting the lab materials as resource materials, available 24/7, to be used when needed. This suggests a just-in-time approach to learning which increasingly reflects the operation of the commercial world these students will soon enter.

(4) The conversion to the virtual lab is on-going and will be completed by fall 2004. The new challenge will be how to provide staff support to students on the same 24/7 utilization schedule as lab availability.

ACKNOWLEDGMENT

The author gratefully acknowledges the NSF DUE CCLI program, without whose financial support this development would not have been possible; Andrew Snyder, Tim Mulholland, and Logan McKinley who wrote the HTML, CSS, SQL, and ASP module and supervised the first hands-on lab; Chris Adams and Sean O’Connor who wrote the PHP material; Rahul Gupta and Kevin Thomas who wrote the new C# and XML labs; and the University of Virginia’s Teaching + Technology program that is financing the course conversion to the virtual lab concept.

REFERENCES

JavaScript Intermediate Exercise 3 – Radio Buttons

Overview
JavaScript can interrogate the data structure that represents a set of radio buttons in a form. For example, in this exercise we ask the user to select his favorite computer science topic from a list of five. Your code first checks to verify that some radio button was selected; if not, nothing happens. If a radio button was selected, then the value associated with that button is sent as an email.

Sample Code
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Exercise Intermediate 3: Radio Buttons -->
<html>
<head>
<title>Control</title>
<script language="JavaScript">
function validate() {
    // your code goes here
}
</script>
</head>
<body>
<h2>Computer Science Topic Survey</h2>
<form action="mailto:yourname@virginia.edu" method="POST" enctype="text/plain" name="cstopics" onsubmit="return validate();">
    Which of the following is your favorite CS topic?
    <input type="radio" name="favorite" value="theory"> Theory of Computation
    <input type="radio" name="favorite" value="OS"> Operating Systems
    <input type="radio" name="favorite" value="algorithms"> Algorithms
    <input type="radio" name="favorite" value="programming"> Programming
    <input type="radio" name="favorite" value="EC"> Electronic Commerce
</form>
</body>
</html>
```

Exercise
1. This code creates a form that asks the user a question and provides five possible answers in the form of radio buttons. By virtue of the nature of radio buttons, only one selection can be checked – but the user could fail to select any answer before clicking submit. Write the function validate() that can identify this special case of the user having made no selection of any radio button.
2. Change "yourmail" to be your email ID.
3. In function validate(), detect whether any radio button has been selected. If so, validate() returns true, otherwise it returns false. If the returned value is true, the HTML form will attempt to send the selected value via email; if false, no email is sent.
<table>
<thead>
<tr>
<th>LAB TOPIC</th>
<th>NBR OF EXERCISES</th>
<th>LAB EXERCISES</th>
</tr>
</thead>
<tbody>
<tr>
<td>HyperText Markup Language (HTML)</td>
<td>12</td>
<td><strong>Beginner</strong>: structural setup; page layout; text manipulation; special characters; images; <strong>Intermediate</strong>: image maps; tables; frames. <strong>Advanced</strong>: forms; meta tags; web forms.</td>
</tr>
<tr>
<td>Cascading Style Sheets (CSS)</td>
<td>12</td>
<td><strong>Beginner</strong>: embedding/linking; HTML element selectors; classes; ID selectors. <strong>Intermediate</strong>: text manipulation; background; borders and spacing; layout; context selectors and grouping. <strong>Advanced</strong>: pseudo-classes; pseudo-elements.</td>
</tr>
<tr>
<td>JavaScript</td>
<td>15</td>
<td><strong>Beginner</strong>: writing your first script; creating HTML tags; user input and output; loops and tables; payroll calculator. <strong>Intermediate</strong>: forms and text fields; validating an email address; radio buttons; check boxes; self-grading tests. <strong>Advanced</strong>: image rollovers; slide shows; real-time clock; controllable clock; working with cookies.</td>
</tr>
<tr>
<td>Perl/CGI</td>
<td>10</td>
<td><strong>Beginner</strong>: sample Perl operations; random numbers; lists; dealing four poker hands; time manipulation; subroutines. <strong>Intermediate</strong>: hash tables; files; string matching. <strong>Advanced</strong>: CGI; registration lists; surveys.</td>
</tr>
<tr>
<td>SQL and regular expressions</td>
<td>24</td>
<td><strong>Regular expressions</strong>: basics; repeating; positioning. <strong>Beginner</strong>: select; where; order by; insert; update; delete. <strong>Intermediate</strong>: like; between; in; distinct; group by; aliases; aggregate functions; create table; alter table; drop table. <strong>Advanced</strong>: nested selects; SoundEx; join; deterministic functions; non-deterministic functions.</td>
</tr>
<tr>
<td>ASP</td>
<td>25</td>
<td><strong>Beginner</strong>: structural setup; response.write; retrieving from forms; retrieving from querystring; variables; control constructs; subroutines and functions; session state; application variables; server variables; debugging. <strong>Intermediate</strong>: reading and writing cookies; server-side includes; response object methods; VBScript functions; error handling; debugging. <strong>Advanced</strong>: browser details; CDONTS; files; output from a recordset; global.asa; setup instructions for using IIS and ASP.</td>
</tr>
<tr>
<td>Flash</td>
<td>3</td>
<td>Create Flash movies of moving and interactive objects.</td>
</tr>
<tr>
<td>PHP</td>
<td>15</td>
<td><strong>Beginner</strong>: Structure, echo, variables, forms, strings, functions. <strong>Intermediate</strong>: control structures, arrays, cookies, session variables, sending email. <strong>Advanced</strong>: files, directories, database setup, database manipulation.</td>
</tr>
<tr>
<td>C#</td>
<td>9</td>
<td><strong>Beginner</strong>: Visual Studio .NET; GUI elements and form designer; common syntax. <strong>Intermediate</strong>: ASP .NET web forms and SQL; MySQL databases using ADO .NET; integrating C# with Excel. <strong>Advanced</strong>: creating an XML web service using C#; language interoperability using C# and web services; dynamic control creation in web applications using C#.</td>
</tr>
<tr>
<td>XML</td>
<td>16</td>
<td><strong>Beginner</strong>: creating an XML document; XML parsers; advanced components; attributes vs. elements; displaying XML. <strong>Intermediate</strong>: legal building blocks; advanced DTD; linking DTD and XML; W3 XML schemas; advanced schemas. <strong>Advanced</strong>: extensible stylesheet language (XSLT); simple API for XML parsing; W3C document object model; simple object access protocol (SOAP); introduction to XML web services.</td>
</tr>
</tbody>
</table>