

WRITING IN XHTML: AN ALTERNATE INTRODUCTORY COURSE TO COMPUTER SCIENCE

Mech Johnson
Department of Computer Science
UW-Washington County
400 University Drive
West Bend, WI 53095
mejohngo@uwc.edu

Abstract

Freshmen arriving in college are often uncertain what they want to study. A student tentatively interested in Computer Science (CS) might look for a Computer Science course less challenging than CS I and II, an option which is offered at many colleges and is intended for Computer Science majors. Many colleges do offer an introductory computer concepts course, here referred to as a CS 0 course, which does not carry CS major credits, but is intended to introduce a student to Computer Science. In this paper, we examine what type of course is typically used for a CS 0 course, and contend that a course in web page creation using XHTML is a better way of introducing students to Computer Science than those courses traditionally offered as CS 0.

1. AN ACCIDENTAL DISCOVERY

The Computer Science department of the thirteen two-year Colleges of the University of Wisconsin system offers an Introduction to Computer Science course, here called CS 0, and a Computer Science I and II programming sequence, also referred to as Programming 1 and 2 [1]. In an effort to update the CS curriculum, and counteract the decreasing enrollment in the programming courses, we designed a 2-credit course in Web Page Development, which introduces students to XML, and uses XHTML and Cascading Style Sheets (CSS) to develop web pages. The success of this course has been astonishing, not only because of its high enrollment, but also because of student and teacher satisfaction with the course. When at the end of the course several students expressed interest in the next CS course, it occurred to me that the Web Page Development course might serve as a better introduction to Computer Science than CS 0.

To support the thesis that web creation using XHTML is indeed a good way of introducing students to Computer Science, we first look at a typical CS 0 course for purposes of comparison, then we explore briefly how web development technologies usually fit into a CS curriculum, and finally, we examine the Web Page Development course itself, and how it is a good way of introducing students to Computer Science.

2. TRADITIONAL CS 0

In the UW Colleges, Introduction to Computer Science is a 3-credit elective course which offers two components: 1) Computer Fundamentals (1 credit) and 2) Introduction to Programming (2 credits). This course has no prerequisites. The Computer Science I and II sequence has the prerequisites of previous programming experience and competency in college algebra; it takes a relatively good student to be able to start his/her CS studies with the CS I, II series. More typically, a student will start with CS 0, Introduction to Computer Science. The CS 0 course offered at the UW Colleges is examined here not for the reason that it lays claim to be a template CS 0 course, but rather for the purpose of placing it in juxtaposition to the Web Page Development course first introduced in the UW Colleges in fall 2004. CS 0 is examined by looking separately at its two components, Computer Fundamentals and Introduction to Programming.

2.1 Computer Fundamentals in a CS 0 Course

The computer fundamentals part of the Introduction to Computer Science offers “An overview of computers, what they are and how they work [2].” Computers are typically discussed by considering the five major functions of a computer: input, output, processing, storage, and communication [3]. A chapter on the Internet is included and important topics such as binary numbers and Boolean logic operators are also mentioned. This 1-credit part of the CS 0 course certainly has validity, but for a student truly interested in testing the waters of Computer Science, this theoretical part is not very inspiring, since it gives the student little idea of the process of programming.

2.2 Introduction to Programming in a CS 0 Course

The 2-credit programming part of the CS 0 course has undergone many changes over the past 15 years. For a while, the programming was done in QBasic or Pascal, later it was done in Visual Basic, and more recently it is done in Visual Basic .NET [4]. One can argue that as long as QBasic or Pascal was used, a good educational introduction to programming was truly possible, because there was little overhead in learning the minimal development environment before writing the first program. However, with Visual Basic .NET the Integrated Development Environment (IDE) poses not only a time-consuming but also a confusing factor for the students when learning how to program. Oddly, the students tend to enjoy learning the IDE and assembling forms with familiar clicking and dragging, and they gladly follow the directions for typing in a given code segment, but it is when the teacher steps back and asks the students to think up their own code that frustration sets in. The boundary between what the IDE provides and the code the students need to create themselves has become blurred enough to confuse and frustrate the students.

2.3 Comments on the Current CS 0 Course

After teaching CS 0 for many years, I have become increasingly dissatisfied with the course. The very topic of “computers, what they are and how they work” does not lend itself well to a 1-credit survey course, nor does 2 credits in Visual Basic .NET allow for a good introduction to the discipline of programming. In both cases, the very scope or open-endedness of the topics makes it a difficult course to teach well. Erring on the side of too small a scope, making the course accessible, leads to the course not being helpful at all, while erring on the side of too broad a scope makes the students resentful over the difficulties encountered in an elective introductory course. In either case, the course has not been very inspiring for students.

3. CHOOSING PUSHDOWN AUTOMATA OVER TURING MACHINES

When designing the course for web page development, the motivation was not to offer an alternate approach to introducing Computer Science, nor was it a conscious decision that if programming was too difficult or too confusing to learn within an introductory CS course, then perhaps learning a simpler system first would be a more appropriate approach. Yet, in hindsight, it is interesting to argue that it makes more sense to start with learning a Context Free Language such as XHTML [5] than it does to start with learning a Recursively Enumerable Language such as Visual Basic .NET, just as it makes more sense to study a Pushdown Automata before studying a Turing Machine [6]. Where the open-ended scope of the traditional CS 0 course can easily lead to dissatisfaction in student and teacher alike, the very simplicity of writing in XHTML makes it a satisfying

2-credit introductory course to Computer Science. Before we examine the 2-credit Web Page Development course as an introductory course to Computer Science, let us briefly examine where web page development usually fits in a college curriculum.

4.TEACHING WEB DEVELOPMENT TECHNOLOGIES

A good treatment of how web technologies can be integrated in Computer Science can be found in *Teaching web development technologies: Past, present, and (near) future* [7] by Billy B L Lim. His suggested course, similar to courses which can be found in the catalogues of many colleges, comes with a hefty list of prerequisites and is intended for juniors, seniors and graduate students. However, an advanced course in web development technologies does not diminish the argument for an introductory course in XHTML. In fact, just the opposite, an advanced course in web development increases the usefulness of the introductory XHTML course for Freshmen, since the knowledge gained in the Freshman XHTML course is applicable not only within its narrow scope of a 2-credit introductory course, but also within the context of serving as a foundation to a more advanced course. It is the limited scope of the introductory Web Page Development course that we examine next.

5.THE WEB PAGE DEVELOPMENT COURSE

5.1Topics Covered

Following is a list of topics typically covered in the Web Page Development course [8], [9], which carries 2 credits and has no prerequisites:

- The Internet and WWW (Overview)
- Extensible Markup Language (XML)
- Well-Formed vs. Valid Documents
- HTML vs. XHTML
- Lists
- Hyperlinks
- Internal Styles
- External Cascading Style Sheets (CSS)
- Images, Image Maps
- Multimedia
- Tables, Nested Tables
- Forms

5.2Course Management

Each topic is connected with an assignment that typically involves publishing a web page on the university server. Notepad is used to write the XHTML and CSS code. The students are required to use DTD XHTML 1.0 Strict, and each page must be validated using the W3C Markup Validation Service [10]. The course concludes with a final project consisting of a multi-page web site created by the student.

5.3 Remarks on the Content

It is not within the scope of this paper to discuss the entire content of the introductory Web Page Development course. However, some selective comments are warranted in order to show the approach that is emphasized in the course.

5.3.1 Markup

Students often have some background in assembling semi-structured documents in HTML, but that misses the point of the course. An important part of the course is to look at documents in terms of data (content), structure and presentation. The origin of the term “markup” from within the typesetting process leads to the discussion of the Standardized General Markup Language (SGML) as a standard for descriptive markup.

5.3.2 XML

XML is presented to the student as an application of SGML, and the students are asked to write a small XML document. This topic is usually of interest to students since many have previously encountered the acronym XML afloat among all other acronyms. Bill Gates’ open letter in Microsoft .NET Today is interesting supplemental reading on the significance of XML [11] and foreshadows Microsoft’s recent announcement that starting in 2006, Microsoft Office (code-named Office 12) will save most files in the Extensible Markup Language. [12]

5.3.3 HTML vs XHTML

XHTML is an application of XML, and is thereby more rigorous than HTML, a language which has mainly served as a presentation language with confusing options of browser-specific tags, and without a distinction between the data, structure, and representation of a document. In contrast to semi-structured HTML, XHTML is a strictly-defined Context Free Language, and the rigor that needs to be adopted in XHTML, though simple, is a good introduction to what the student will encounter later in true programming languages.

5.3.4CSS

Cascading Style Sheets are introduced early, because they add clarity to the distinction between the content and representation of a document. A common exercise assigned when discussing this topic is to write two different versions of style sheets for a single web page, and thus have the students convince themselves of how much a document can change by altering only the external .css file. Just as validation is required for the XHTML code, validation of the external CSS file is required to make sure that the rules of representation are observed [13].

5.3.5Forms

The emphasis when covering forms is merely on how to write the XHTML code for assembling the form. It is a good topic with which to conclude the course because one can now make the student aware of the limitations of static web pages, and can explore in an intuitive way the concepts of client and server side programming.

6.6. EXPERIENCE

Voicing dissatisfaction with the traditional CS 0 course has become not uncommon among the faculty. It is difficult to get students interested in an overview of computers, and the introduction to programming using Visual Basic .NET either deteriorates into assembling forms doing not very interesting things if too little coding is requested, or, if too much coding is requested, the student becomes frustrated, lost in the intricacies of the IDE and the seemingly limitless scope of the language itself, a problem that surfaced when languages such as QBasic or Pascal were dismissed as being too old-fashioned. Though the Web Page Development course was not designed as an introduction to Computer Science, the very success of the course has prompted an examination of the reasons for the success - reasons which are now explored, and have lead me to conclude that this Web Page Development course serves as a better introduction to CS than the traditional CS 0.

6.1Computer Fundamentals in Web Page Development

The topics of the Web Page Development course do not overlap significantly with the Computer Fundamentals offered in the CS 0 courses. Indeed, if any type of learning applicable to the Computer Fundamentals topic could claim to have the same impact on the student as the active learning that takes place when writing in XHTML, then I would not be promoting a Web Page Development course over a standard CS 0 course. Though the teacher will have the feeling that a lot of material gets covered in the Computer Fundamentals section of the CS 0 course, the student will often comment, “boring”, “did not learn a lot”, and this might well be true from the perspective of the student. The nice thing about the Web Page Development class is that it involves active learning, and

though it will cover fewer Computer Fundamentals topics, the knowledge gained has a deeper impact on the student. When teaching Web Page Development, one begins to suspect that the students in the course gain a better “overview of computers, what they are and how they work, [2]” than do the students in CS 0 for whom that learning goal was defined.

6.2 Introduction to Programming in Web Page Development

XHTML and CSS are not programming languages, but they are a good way of preparing students for learning how to program. Next, we discuss several skills that are developed in the Web Page Development course, skills which are useful for the students to acquire before learning how to program.

6.2.1 Writing, Validating and Running Code

The IDE for writing XHTML and CSS is Notepad, which not only causes no overhead in learning, but is also a pleasing start to demystifying the very process of coding. The students are encouraged to write, validate, and then run the code in the browser. Clearly, validation is not compilation, but validation does serve a similar purpose in the write-validate-run sequence; that is, syntax errors are flagged.

6.2.2A Precise Science

In a programming course, students often stumble over the mere fact that the code needs to be written precisely. This skill is acquired necessarily by a student writing XHTML in Notepad. For example, XHTML is case sensitive, and all nonempty elements require end tags. Though the concepts are simple, the lesson eventually is well-learned: the computer is a precise, digital system, and not an analog system tolerant of small errors.

6.2.3 Commenting Code

Commenting code is a good practice, as is commenting the entire file with a header to state the name, date and purpose of the web page or program. These commenting practices are applicable not only in XHTML and CSS, but also in full programming languages such as C++, Java, and Visual Basic .NET.

6.2.4 Debugging

The virtue of validating often holds true for writing an XHTML document just as the virtue of compiling often holds true for programming. An XHTML document that

produces dozens of errors during validation is best debugged by commenting out major portions of the code in order to get at the source of individual errors. Just as in programming, the students learn quickly that the line flagged for an error is not necessarily the line that has caused the problem.

6.2.5 Syntax and Logic Errors

It tends to come as a surprise to the student when an XHTML document completely validates, but the web page displayed is not what the student expected, or what the assignment required. This leads to an opportune time to talk about the difference between syntax and logic errors.

6.2.6A Sense of Accomplishment

More arguments could be enumerated to support the fact that learning a Context-Free Language is a good preparation for learning how to program. Instead, I choose as a last example an argument of a less precise nature. Any programmer knows the sense of accomplishment after having created a program that works. However, introductory programs tend to be rather simple, and are not necessarily something that can be easily shared with friends. I have observed that a driving force in the Web Page Development course is that the output, that is, the displayed web pages, are a compelling goal for the students to strive for, a rewarding sense of accomplishment that can easily be shared with other people.

7. CONCLUSION

After years of teaching Introduction for Computer Science, an elective 3 credit course, I designed and began to teach a 2-credit course in Web Page Development with no prerequisites, a course wherein students write valid XHTML documents and CSS using Notepad. This course has turned out not only to be a successful course in terms of enrollment and good evaluations from students, but it has also lead me to conclude that writing in XHTML is a preferable way of getting students interested in Computer Science over the traditional CS 0 course offered.

8.REFERENCES

- [1] Campbell, R.D, Cover, C. F., Hawthorne, E.K., Klee, K. J., Guidelines for Associate-Degree Programs, Information systems, 2004, ACM, ISBN: 1-58113-889-X
- [2] University of Wisconsin Colleges, Catalogue 2005-2007, <http://www.uwc.edu/academics/catalog.pdf>
- [3] Pfaffenberger, B., Computers In Your Future 2004, Brief, 6/E, Prentice Hall, 2004, ISBN: 0-13-140452-0
- [4] Salvage, J., Visual Basic .NET Coach, Addison-Wesley, 2003, ISBN: 0-321-11350
- [5] Pullum, G., Context Free Grammars and Languages, <http://www.kornai.com/MatLing/cflfinal.pdf>
- [6] Sipser, M., Introduction to the Theory of Computation, PWS Publishing Company, 1997, ISBN: 0-534-94728-X
- [7] Lim, Billy B.L., “Teaching web development technologies: Past, present, and (near) future”, *Journal of Information Systems Education* (2002) http://www.findarticles.com/p/articles/mi_qa4041/is_200201/ai_n9068437
- [8] Musciano C., Kennedy B., HTML & XHTML, The Definite Guide, O’Reilly, 2002, ISBN: 0-596-00382-x
- [9] Hughes, C.M., The Web Wizard’s Guide to XHTML, Addison-Wesley, 2004, ISBN: 0-321-17868-8
- [10] W3C, Markup Validation Service v0.6.7 <http://validator.w3.org/>
- [11] Gates, W., Microsoft NET Today, an open letter to the Developers & IT Professionals, June 14, 2001 http://download.microsoft.com/download/a/f/4/af49b391-086e-4aa2-a84b-ef6d916b2f08/net_today.pdf
- [12] Ricadela, A, “Microsoft To Make Office Files More Internet-Friendly”, *Information Week*, June 2, 2005 <http://www.informationweek.com/story/showArticle.jhtml?articleID=163702927>
- [13] W3C, CSS Validation Service, <http://jigsaw.w3.org/css-validator/>