

# Delivering Internet and Programming Courses Online

Barbara Mento<sup>1</sup>  
Donna Tupper<sup>2</sup>  
Kathleen Harmeyer<sup>3</sup>  
Sylvia Sorkin<sup>4</sup>

Mathematics, Computer Science, Internet and Multimedia Technology Department  
Community College of Baltimore County - Essex Campus  
Baltimore, MD, 21237, USA

## Abstract

A new thirty-credit certificate program in Internet and Multimedia Technology (I/MMT) has been designed at the Essex Campus of the Community College of Baltimore County (CCBC-Essex). This paper provides descriptions of three required courses in the program for which online versions have been created: Internet Literacy, Visual Basic I and Internet Programming.

**Keywords:** Distance education, Internet literacy, Internet programming, JavaScript, Visual Basic, Authorware

## 1. I/MMT CERTIFICATE PROGRAM

Employers in business, entertainment, and education seek employees who can help them keep pace with the demands of global communication. To prepare students for jobs in web-based interactive technologies, CCBC-Essex has designed and implemented a thirty-credit I/MMT program. An associate degree program to prepare students for entry-level employment, and to provide opportunities for upgrading technical skills, and for further education through articulated programs, is now being designed. Students in the program choose one of two options: Internet Specialist or Multimedia Specialist (Sorkin 2000).

Students in the Multimedia Specialist option learn to design and develop interactive computer materials for advertising, publishing, animation, and education. Students learn computer screen design techniques with courses in art and digital imaging. Students in this option do not have to be programmers; they learn to produce interactive programs without programming by using authoring and animation software. Students in the Internet Specialist option learn to produce dynamic web sites with animation, interactivity, and database access to prepare them for Internet marketing and web-site design and administration. **Figure 1** lists suggested courses by semester for this option.

## 2. INTERNET LITERACY COURSE

In the Internet Literacy course, we reach a varied audience, in part because the course also satisfies a general education requirement. There are students who are very interested in learning how to create web pages and those who are looking for a required technology credit. We serve from those who do not own a computer to those who already have web sites up and running. However wide a range of experiences our students bring to the course, we notice that no one knows all of the content for this course in advance. For example, owing perhaps to the eclectic nature of their learning about the web, many students think they know very much about searching and surfing. Yet even the most "experienced" are interested to learn the Boolean techniques and natural language processing services available on the web, or that not all search engines operate identically. To meet the needs of such a diverse group, we invented a few activities that stimulate thought and promote further exploration of the web.

One of the more difficult concepts for students who are not computer science or computer information systems majors to master is that of packet switching. The concept of a text file and 16K blocks of data are foreign notions. We use the following exercise to illustrate it. The day that packet switching is introduced, a two-page lab is conducted. Students hand in their pages and the papers are scored and handed back at the beginning of the next class in the following manner: I greet the students and tell them that I will be handing back their labs. I go to one student and ask her if she is busy. She says "No," so I hand her the first page on the stack of papers and ask her to deliver it to the proper owner.

---

<sup>1</sup> bmento@ccbc.cc.md.us

<sup>2</sup> dtupper@ccbc.cc.md.us

<sup>3</sup> kharmeyer@ccbc.cc.md.us

<sup>4</sup> ssorkin@ccbc.cc.md.us

Then for the next student I repeat the process with page 2, continuing to other students. If I encounter a student wandering around the room looking for the owner to a paper I acknowledge that he is busy and go on to another student for the next page. They are temporarily bewildered but follow instructions. And soon all papers are in the correct hands. As they settle back into their seats, perhaps wondering what kind of a ditz they have for a professor, I ask if everyone has his or her own pages back.

I ask in which order they were received. Some receive page 2 then 1, others 1 then 2. Then I say "Welcome to Packet Switching." They all laugh and understand. It is not unusual for them to note this on the course evaluations as one of the best activities of the semester.

Currently one section of the course is taught online. A recent study commissioned by National Education Association and Blackboard Inc. includes 24 quality benchmarks for Internet-based distance learning in higher education. One of these benchmarks is, "Student interaction with faculty and other students is an essential characteristic and is facilitated through a variety of ways, including voice-mail and/or e-mail" (NEA 2000). Chat sessions can be helpful if a few rules of the road are instituted. The students who are experienced with chat approach it in a relatively freewheeling manner, interrupting, changing the subject, and addressing individual students. Since I am old school and prefer a more controlled environment, I promote "bonding" for a few minutes at the beginning of the lesson and then ask "OK, are we ready to get started?" As they all answer in their own affirmative way, the lesson gets underway. To manage interactions, students enter a question mark (?) to indicate they have a question, and enter an exclamation point (!) to show a desire to make a comment. Since the student's name precedes the interaction character on the chat screen I am able to "call" on the student who then has the floor. Other students respect who has the floor and wait patiently for the question or comment to appear on the screen. On occasion this takes a long time as the student types a lengthy question or comment. However, we are able to provide a moment for students to interact during the lesson before the moment gets away from them.

Most of our students in the online section have Windows computers. We have used the free Microsoft conferencing software, *NetMeeting*, to great advantage. The instructor has a web camera and students who join the meeting are able to see her image as she conducts a synchronous activity. Students use headphones and microphones to speak to each other over the Internet. The major benefit of *NetMeeting* is the ability to share resources. The desktop is sharable so that the instructor can work at her computer while the students can see and interact with her programs. In addition, there is a chat window, as well as a whiteboard on which students may draw

<b>Internet Specialist Option</b>		
<b>Semester 1</b>		<b>Credits</b>
ENGL 101	Composition	3
CMSC 155	Internet Literacy	3
MULT 109	Multimedia Authoring I	3
ARTS 102	Two Dimensional Design	3
<b>Semester 2</b>		
CMSC 157	Internet Programming	3
CMSC 107 or CMSC 225 or CMSC 201	Visual Basic I or C++ Programming or Computer Science I	3 or 4 or 4
2 Electives		6
<b>Semester 3</b>		
CMSC 235	Java Programming	3
MULT 221	Project Development	3
<b>Total Credits:</b>		<b>30</b>
<b>Electives</b>		
ENGL 213	Technical Writing	3
CMSC 233	Internet Database Programming	3
CMSC 142	Unix Operating System	3
MULT 121	Digital Imaging	3
MULT 201	2D Animation	3
MULT 205	3D Advanced Animation	3
MULT 209	Multimedia Authoring II	3
MULT 211	Digital Audio/Video	3
MULT 231	Portfolio/Coop	3

**Figure 1. Suggested courses by semester for Internet Specialist option.**

and type text. Students without a Windows system may use the campus computer labs to participate or may run a PC emulator to participate on their own systems.

The amount of conversation and transmission of content is greatly increased by this tool and students are delighted with the technology. The future of the camera-based lesson is in the use of the Microsoft Office 2000 products. According to the Microsoft website, *PowerPoint 2000* provides a window for conferencing within the presentation window. An information page detailing how to "Broadcast Your Presentation Live on the Internet" can be found at <http://www.microsoft.com/office/powerpoint/Manage.htm>.

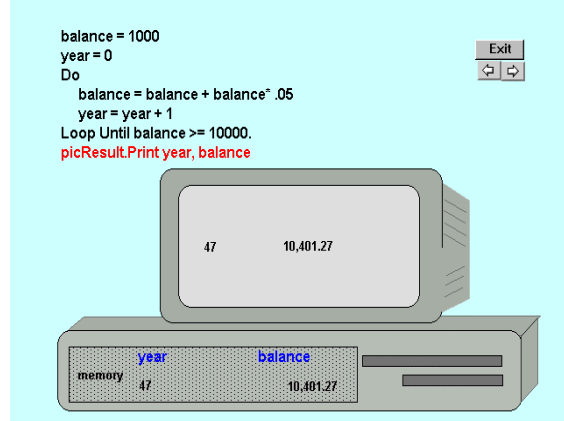
### 3. ONLINE VISUAL BASIC PROGRAMMING COURSE

An online course has been developed to teach Visual Basic Programming to students who have had no previous programming experience (Mento 2000). Many of these students are taking a computer science course because they have heard that computer science is a lucrative field with great employment opportunities. However, they have no background to help them understand the logic involved in computer programming. Sensing that some students became discouraged quickly because the abstract concepts were difficult to grasp, the first author started using multimedia in the computer science classroom five years ago. She found that using visuals and animation helped students understand and remember programming concepts.

Macromedia *Authorware* was selected for development of interactive training materials for the classroom and for the online course. It was chosen, rather than HTML or Java to create animation, because of its ease of use and easy conversion to the web. Students must download free plug-ins for *Internet Explorer* from <http://www.macromedia.com> and for *Netscape* from <http://www.Netscape.com>. Future browser versions will have "Shockwave" capability built-in. The metaphor used by *Authorware* is the flowline. The developer adds pre-programmed icons to the flowline to create a lesson. The display icon is used to place text or images on the screen. Other icons are used to control timing, erase previous screens, create interactivity, control flow, make decisions, and use variables and functions.

The interface can be designed in many ways using the framework icon. The more intuitive the interface, the less frustrating it will be for the student. The interface shown in **Figure 2** uses only three buttons: Forward, Back, and Exit.

This interface was tested by students in a campus-based Visual Basic course prior to using it for the online course, and has the advantage that the student can stop



**Figure 2. Multimedia demonstration of looping for an online Visual Basic course.**

at any time with the Exit button or can repeat the previous pages, animations, or questions as desired.

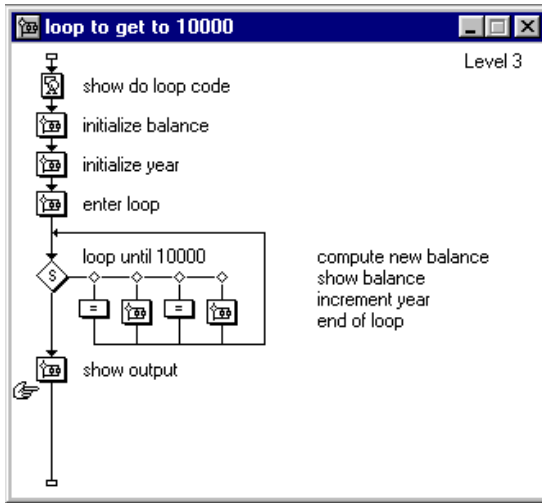
This is especially helpful for the student who needs to see content repeatedly. *Authorware* has simple drawing tools to assist with creating visuals for the student. For more sophisticated drawing capability, one would use a more extensive drawing program to create images, then import them into the *Authorware* project. File size is the drawback to using clip art or sophisticated drawings in an online course: as the size of the file increases, the download time increases. The instructor must weigh the advantages to the student in learning value. For instance, to show the Visual Basic interface itself, it is important to import an exact screen capture, but to display the monitor output, it suffices to use a "homemade" simplified drawing of a computer monitor.

Animation is used early in the programming class to demonstrate the concept of data "moving, but staying" as input from files to memory, and from memory to output. This helps the student understand and remember the visual image of "where" the data is at all times. Beginning programming students need help understanding when data is overwritten and when it is merely used for output. Animation is created in *Authorware* by adding an icon and designing where the image should start and stop its movement. Duplicate images are created to show data moving, but staying at the same time.

**Figure 2** shows part of a lesson demonstrating looping. To highlight individual statements in the order that they are executed, *Authorware's* scripting capabilities were utilized. The problem illustrated generally captures students' interest: "How many years will it take to accumulate \$10,000 if you start with \$1000 and earn 5% interest compounded annually?" This visualization enables the student to watch as the computer works through the Do loop, statement by statement, and at the

same time see how the variable contents change in memory. **Figure 3** shows the *Authorware* icons used to create this "visual" loop.

Creating interactivity on the web can be cumbersome and server-dependent in other web languages, but *Authorware* has a simple device for creating this interactivity. Types of questions can be multiple choice or fill-in-the-blank. Fill-in-the-blank is particularly powerful because the instructor can choose multiple correct answers (e.g., 5 or 5.0). The instructor can give the student feedback for correct and wrong answers, and can give hints for nearly- correct answers. The student can also be allowed to try again, and be provided with the answer after a few incorrect tries.



**Figure 3.** *Authorware* icons used to create a multimedia demonstration of looping.

Using multimedia in the online Visual Basic course has helped produce lessons that are closer to classroom lectures than to the textbook. Classroom lectures are full of animated gestures of the teacher, the use of overheads and videos, and interaction between students and teacher. All those components are multimedia features of the course.

#### 4. INTRODUCTORY INTERNET PROGRAMMING COURSE

Internet Programming is a comprehensive course in web design and development. Principal topics of the course are advanced HTML and JavaScript. Prior programming experience is NOT required. However, some HTML experience is a prerequisite. The prerequisite for this course is the Internet Literacy course, which includes an introductory amount of HTML. Although first offered in the traditional

classroom setting, an online version of the course has been developed (Tupper 2000).

Students taking a computer course tend to be comfortable with computers, therefore the procedure for logging into the class, using chat rooms and bulletin boards and sending email with attachments are nothing new to them. What is new to them is taking a course online. It is critical that the instructor advises students at the start of the semester about the course content, procedure, assignments and examination policy. A clear calendar of daily or weekly activities helps students manage their time. This is a skill that is not inherent to most students. Having students turn in weekly activities helps to keep the entire class in the same general location in the course. The class can still be asynchronous, but students tend to not get as far behind as their counterparts in other online classes.

Keeping this course at a level that will benefit everybody is the first obstacle to overcome. At the start of the semester it is important to check each student's background to make sure all students were properly placed into this course. Students enter this class with backgrounds ranging from "surfing the web" to having taken Java. The one constant found in most programming students (and many professional programmers) is lack of documentation and poor organizational skills. Exercises in file hierarchy and insisting lab assignments and homework assignments are kept in separate folders helps students develop organizational skills. Mandating that students comment all of their source code will force a minimal amount of documentation, but also has a hidden advantage. These students are savvy enough to share critical information. Having students describe how their code works forces them to analyze how programs behave.

Another common trait in most students is a lack of knowledge when it comes to issues involving copyright laws, ADA guidelines and fundamental design principles. The Internet is the perfect place to discuss these issues because of the large number of websites devoted to these issues. A list of valuable websites can be found at the end of this paper. A problem inherent to teaching the course in a traditional lab environment is the problem of keeping the students' attention. It is common for a few students to surf the web and check email while attending a traditional lecture. This is very distracting to the instructor as well as to other students in the class. This is not a problem for online classes; in fact it can be beneficial. Students continually email the instructor regarding new websites on HTML, JavaScript, Perl, clip art, and many other course-relevant issues. Critiquing professional websites is another way to channel this energy.

From the instructor's perspective, organization is everything. Before the semester begins it is useful to have a checklist of what needs to be done and when it needs to be done. Not only will this keep the instructor

on track, but it helps the secretaries and testing center personnel be aware of what is happening. Again, the online environment is new to all of us and open communication makes things go a lot smoother. Items to include in the checklist can involve: 1) an introductory letter to help students log into the class, 2) a list of weekly reminders to email students so they keep on track, 3) mid-semester progress reports to be mailed to students, 4) copies of all quizzes/exams and the dates they need to be at the testing facility, and 5) current student contact information. Number 5) is critical but difficult to accomplish. Many students register after initial grade reports are printed. These students may not know to email the instructor upon registration. If the instructor does not frequently obtain a current class list, students can easily be enrolled without ever receiving information about the class. Even in a fifteen-week semester, one lost week is a lot of time.

Once the semester has begun, there are many little things an instructor can do to keep her head above the rising tide of emails. Have students email accounts set up such that their name appears on all incoming emails. It can be very difficult to track 100 students by email name alone, and when you have three Steve's in one class, it saves you from checking your email address book each time an assignment is submitted. Keeping a disk with a folder for each student's assignments is a must. Students will modify programs many times and it helps for one of you to have an original copy. It also gives you ready access to all students' assignments as well as any images they may have used. Try to convince students to use small images. They really don't understand the pain involved in a long download.

We cannot guarantee what software is available to students taking the course online, and did not want to require students to buy additional software. So the first step in developing an online version of the course was to create an "art gallery" of backgrounds and images that students could use for various topics throughout the course. Using *CorelDraw* and *CorelPaint*, the instructor created GIF and JPG files for student use. Although encouraged to create their own graphics, students are also allowed to use public domain sites and adhere to copyright laws.

Students who had taken the course in a traditional classroom environment came into the course with a wide variety of backgrounds and abilities. The course developer realized that students need various, alternate sources of reference material on HTML, JavaScript and design principles. To challenge students of all levels, a list of recommended resources (both textual and web-based) for students was compiled. To keep students on task in an asynchronous course environment, lecture notes for the online course are grouped by week. Students are expected to spend three hours per week on lecture notes, and another hour per week on lab assignments. Since many students are new to taking courses online, practice quizzes were developed and put

Assignment 10: Even/Odd Example
<p>Type in the code below and answer the questions in the commented sections.</p> <pre> &lt;HTML&gt; &lt;!--This program will determine whether a number entered is even or odd. Notice a few new items. I used the modulo operator % and the if- else control structure. Also, notice that this function required a parameter. ---&gt; &lt;HEAD&gt; &lt;SCRIPT LANGUAGE="JAVASCRIPT"&gt; function conditional(numberEntered){ &lt;!--If the number entered, when divided by 2, has a remainder of 0 (in other words, if it is even).---&gt; if ((numberEntered % 2) == 0) { &lt;!--What does the following code do. Write a detailed sentence, describing the code.---&gt; result = "&lt;HEAD&gt;&lt;TITLE&gt;Tickle Me Pink&lt;/TITLE&gt;&lt;/HEAD&gt;" + "&lt;BODY BGCOLOR=#FF0080' TEXT=#FFFFFF' &gt;&lt;P&gt;" + numberEntered + " is an even number"; newWindow = window.open("", "", ,width=250, height=5,scrollbars=yes"); newWindow.document.write (result); newWindow.document.close(); } &lt;!--If the code is not even, in other words it is odd ---&gt; else{ result = "&lt;HEAD&gt;&lt;TITLE&gt;Green With Envy&lt;/TITLE&gt;&lt;/HEAD&gt;" + "&lt;BODY BGCOLOR=#008000' TEXT=#FFFFFF' &gt;&lt;P&gt;" + numberEntered + " is an odd number"; newWindow = window.open("", "", ,width=250, height=5,scrollbars=yes"); newWindow.document.write (result); newWindow.document.close(); } } &lt;/SCRIPT&gt; &lt;/HEAD&gt; &lt;BODY BGCOLOR="#EAF4FF" &gt; &lt;FORM NAME=evenOdd&gt; Please enter a number: &lt;INPUT TYPE=TEXT NAME="input1"&gt; &lt;P&gt;&lt;BR&gt; &lt;INPUT TYPE="BUTTON" VALUE="Even or Odd" onClick="conditional(evenOdd.input1.value)"&gt; &lt;/FORM&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

**Figure 4. JavaScript function assignment for Internet Programming course.**

online to give students additional practice with various topics. Although students must come to campus to take the graded quizzes and exams, they can email all assignments and practice quizzes to the instructor.

The HTML portion of the course begins with a review of the basic tags, followed by detailed discussion of forms, frames, tables, image maps, and META information. Each major topic is accompanied by a lab exercise. The second half of the course is devoted to JavaScript, beginning with a discussion of the differences between Java and JavaScript, and the advantages of each. Because no prior programming experience is required, students are introduced to the various concepts of programming. JavaScript topics discussed include animation, form handling, advertising banners, and scrolling message bars (Sorkin 2000). An assignment from the course can be found in **Figure 4**.

## 5. CONCLUSION

Over the period of one year, three online courses were created by three faculty for the Internet Specialist option of a new thirty-credit certificate program. Each of these faculty had previously taught the course in a traditional mode. Several of the challenges associated with teaching such courses online have been met by judicious organization of instructional material, use of weekly moderated online chat sessions, and use of Macromedia *Authorware* to animate looping constructs.

## 6. ACKNOWLEDGEMENTS

This material is based upon work supported by the NSF under Grant No. DUE - 99500056.

## 7. REFERENCES

Mento, Barbara, 2000, "CMSC 107 Visual Basic." Online. Internet [July 26, 2000]. WWW: <http://student.ccbc.cc.md.us/courses/cmssc107e/index.html>

National Education Association and Blackboard Inc., 2000, "Quality on the Line: NEA and Blackboard Inc. Study Finds 24 Measures of Quality in Internet-Based Distance Learning." Online. Internet [July 26, 2000]. WWW: <http://www.ihep.com/PR17.html>

Sorkin, Sylvia, Barbara Mento, Donna Tupper, and Kathleen Harmeyer, 2000, "Curriculum Development in Internet and Multimedia Technology." ACM 31st SIGCSE Proceedings, March 8-12, 2000, pp. 50-54.

Tupper. Donna, 2000, "CMSC 157 Internet Programming" Online. Internet [July 26, 2000]. WWW: <http://student.ccbc.cc.md.us/~dtupper/Online157/framepg.htm>