

Developing Follow-on Information Fluency Experiences for Students Completing a Computer Literacy Course

Ken Abernethy, Furman University
ken.abernethy@furman.edu

Bob Shive, Millsaps College
shivera@millsaps.edu

A driving force in the effort to develop a follow-on information fluency experience for students is our assertion that the ability to transform data into actionable information should be a primary outcome of the liberal arts experience. This should be in the context of understanding that information fluency is not a skill obtained nor a destination to reach but a journey which is a lifetime of learning, engaging in ideas, and grappling with an ever-expanding body of knowledge and resources used in working with this knowledge.

Final Report

The liberal arts graduate in the complex environment of the world today must be prepared to take advantage of a variety of opportunities and to address a diverse set of issues. Many such opportunities and issues are influenced by a need to acquire and assess information. Success in the information-driven society will be significantly influenced by the quality of the undergraduate learning experience. Our assertion is that the ability to transform data into actionable information should be a primary outcome of the liberal arts experience. This project has focused on such a desired educational outcome by developing a variety of ideas to be incorporated into a new course which we feel will provide an appropriate developmental framework for students to internalize the ability to work creatively in the information age.

One course cannot provide all the requisite experiences needed. And the desired outcome is actually a moving target, as information technology continues regularly to reinvent itself over ever-shrinking time cycles. So, it is important for students to understand that information fluency is not a skill obtained nor a destination to reach but a journey which is a lifetime of learning, engaging in ideas, and grappling with an ever-expanding body of knowledge and resources used in working with this knowledge. The various course ideas proposed build on an assumed set of computer literacy competencies (a comfort level and basic set of skills in the use of a computer) which could be obtained in a variety of ways, including self-training, pre-college work or college-level computer literacy courses. The student is moved beyond this level to confront the issues of developing a path toward information fluency.

A fundamental assumption of the information fluency initiative in the Associated Colleges of the South has been that information fluency lies somewhere in the confluence of computer literacy, information literacy, and critical thinking skills. Information fluency is a complex compendium of knowledge, nuance, and context, heavily influenced by experience and by technology, both of which are constantly changing. Information cannot be divorced from technology and thus one must be aware of current capabilities both in hardware and in software. Students will have knowledge of many facets of technology and, although technology skills are not the focus of course ideas, some attention will of necessity have to be given to this topic.

This report discusses the evolution of ideas considered in developing a follow-on experience for students who have developed knowledge and skills such as one might obtain by taking a computer literacy course. At the outset the thinking was that a course would be developed and focus on extending the information and technology skills of students to help them evolve in information fluency. It was our opinion that information fluent graduates will be in increasing demand as we move further and faster into the digital age. The combination of traditional liberal arts capabilities and information fluency would foster the development of graduates who are much better prepared for the world in which they will live and work.

This course being designed in the project was not intended to be “the answer” for information fluency. Rather, it was to be one model which was an experiment in the development of information fluency. It was expected that experiences from implementation on two campuses would lead toward continuing refinements each time the course was offered. The expectation was that students who completed the course would have made significant progress in their lifelong journey toward information fluency.

Thus, initial thinking in the project led to the development of an experimental course model blending traditional teaching, self-teaching, teamwork, and collaborative experiences both on and off the home campus. After several refinements, it was determined that the course focus topics would be the digital technologies: bioinformatics, information mining, and computer security. The course was to be called “Future Impact of Digital Technologies”.

The primary design objectives of the course were ones for which the student would:

- ▼ learn basic concepts of selected emerging digital technologies;
- ▼ organize themselves into teams to conduct independent research on the application and future impact of selected technologies;
- ▼ experience the value and power of teamwork in assimilating new information and assessing its impact;
- ▼ employ cutting-edge tools/techniques for managing effective distributed team collaboration;
- ▼ analyze, synthesize, and report on their research findings; and,
- ▼ develop and refine their presentation skills by presenting their findings in a cogent and convincing manner, employing three different methods/media: oral presentations (using presentation software such as PowerPoint), on-line Web site, and a written report.

From the beginning the intention was that this course would be designed, developed, and delivered employing a distributive collaborative model. The range of topics in digital technologies to be covered was admittedly broad. Thus, it was unlikely that any one professor would be fully conversant in all the areas. To meet this challenge, a team approach for instruction was a fundamental assumption used in the design for the course. It was anticipated that the project team of Dr. Abernethy and Dr. Shive would be expanded to assist in course delivery and future development..

During the time that planning for the course was in progress however, the “digital bust” occurred, severely depressing the technology sector, the financial markets, and the economy in general. A significant side effect of this huge downward spiral in the technology sector was a greatly decreased interest by students in the selection of computer science courses in general and a desire to major in computer science in particular.

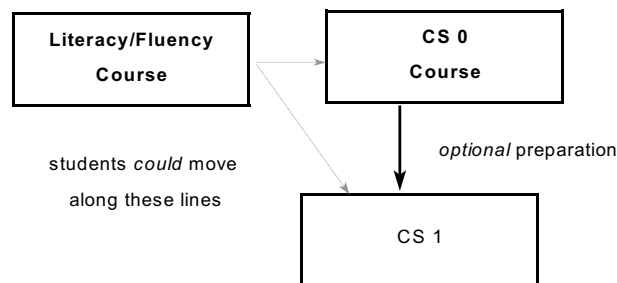
Every academic discipline thinks in the ideal environment but must operate in the real or practical environment. The sudden reduction in students taking computer science courses caused a reevaluation of course offering strategies. In particular, a new departmental focus became that of steering students

toward considering computer science as a major. Offering an experimental course was not a practical strategy. And, our plan for the project was not just to design a course but to implement it and determine whether the design was a sound one and if the course really “worked”.

Thus, the realities of the real-world environment caused a rethinking of offering the course we were designing. There was no less thought that the course being drafted was an excellent idea or that it would not engage students. Rather, the focus became one of departmental need and computer science enrollments. At this point, it became apparent that a better strategy would be to develop a course which would meet aims of the course we had originally designed but that in addition would assist in attracting students to the computer science discipline.

A new idea was a course to bridge the gap between computer literacy and computer science. We felt that such a course could provide an evolutionary step in information fluency for students while at the same time provide an exciting “next step” for students who had completed a literacy course and wanted to learn more about computing. Dr. Abernethy invited his colleague Dr. Kevin Treu of Furman to join in the work to develop a new course as a result of his work with him on the computer literacy course taught at Furman.

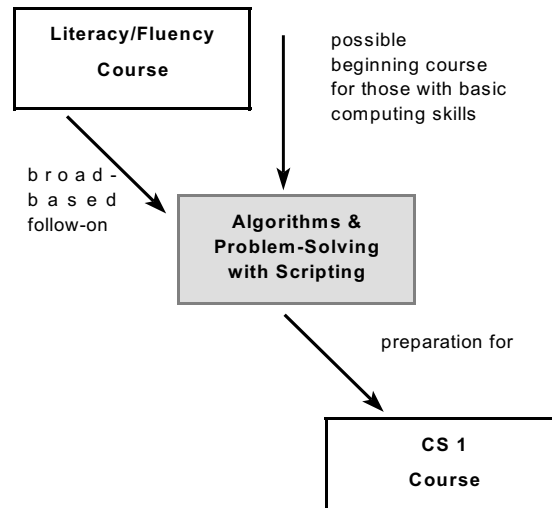
The design of another course was still based on the initial assumption that students had an assumed set of computer literacy competencies obtained in a variety of ways, including self-training, pre-college work or college-level computer literacy courses. Such college-level courses generally focus on the use of computing tools for high-level problem solving and cover topics such as word processing, spreadsheets, databases, and Web authoring using HTML. The typical first “real” computer science course is traditionally referred to as CS1 (in previous curriculum guidelines of the Association for Computing Machinery) and covers algorithmic thinking and programming. Some institutions offer an optional course referred to as CS0 (in the same previous guidelines) and it is typically a gentler introduction to computer science. The diagram below summarizes the relationship among the three courses:



Our experience has shown that few students follow the path from the literacy course to CS1. And, a literacy course usually does not provide significant preparation for CS1. Also, many schools do not offer a CS0 course and thus some students who think they are interested in computing have no opportunity to “dabble” before jumping into the “heavy” beginning computer science course, CS1. This led to two questions:

1. Is there some new course (not literacy or CS0) that could encourage students to take CS1 and at the same time improve their chances of success in CS1?
2. What structure and content of a new course would attract students and serve as a natural follow-on for the literacy course (or its equivalent in preparation) and give a broad-based study of general topics in computing?

These questions led to the development of a course with an emphasis on algorithmic thinking and focused on problem solving using scripting. Such a course could in our estimation provide a desired bridge between the literacy course and the computer science major course CS1 as well as promote information fluency. This is shown by the following diagram:



A problem-oriented approach is a common theme for both the usual literacy course and CS1. As previously mentioned, literacy courses are ordinarily application-oriented using computing concepts as presented through word processing, spreadsheets, presentation software, elementary computer graphics, databases, and Web design with HTML. Furman University has successfully employed an approach that organizes the study of these tools around the common theme of multimedia computing. Millsaps College has been successful in applying many of these tools in its problem-solving literacy course. In both approaches, the covered applications cannot be studied in depth and there is little content that can really be considered to be any type of programming.

The CS1 course focuses on algorithmic problem-solving at a much more detailed level requiring the student to gain proficiency in a programming language such as Java or C++. There is no real focus on high-level applications; rather, the students develop applications or sub-applications as they learn programming.

It can be seen that the literacy course emphasizes high-level problem-solving with existing applications while the CS1 course is grounded in detailed algorithmic problem-solving. And thus, there is a broad expanse between the two courses and hence there is justification for a bridge between the two. The use of scripting for the bridge course would allow for an introduction to algorithmic thinking and programming without being as "heavy-duty" as CS1. This could allow students to get a better flavor of what computer science is and at the same time provide a foundation/preparation for the CS1 course.

From the platform of the literacy course, the bridge course could provide the opportunity for interested students to continue and expand their computing studies through the study of advanced features of applications that they have already learned. From the perspective of the computer science major the new course could be a non-intimidating lower-level introduction to the concepts of algorithmic programming. Students finishing the new course would have a foundation needed to contend with more complex programming ideas.

The implementation of the new course would use scripting in high-level application software. Along with this would be added elements to provide an introduction to the breadth of the computer science discipline. It is our thinking that this approach will address students' curiosity about extended capabilities of applications they have studied and hopefully motivate them to seek further study in the discipline.

The new course initially addresses the basic principles of programming - variables, expressions, control structures, modularity, design, and testing. But, the course is not intended to be a programming course per se. Assignments and projects will be at a lower level of difficulty than those that a student would likely encounter in a CS1 course. However, they should be far more challenging than what is seen in the literacy course.

The course content begins with a topic that students know and are interested in - the World Wide Web. Students will have worked with static Web pages in a literacy course, on their own, or can master an introductory understanding with on-line tutorial material. They begin in the course learning to write Web pages from scratch for both the client-side and server-side environments. Javascript is the language to be used for the client-side scripting and will allow students to quickly create Web pages with interesting and motivating interactive effects. PHP (PHP: Hypertext Preprocessor) will be used as the language for the server-side. Use of this open-source language allows the introduction of the advantages of power and security that are a result of processing scripts before they are sent to a browser. And, since PHP is object-oriented and its syntax is based on that of C++, its use helps to reinforce programming skills learned with Javascript but also provides an introduction to higher-level language syntax.

The next topic to be introduced is advanced database applications and the creation of advanced utilities in Microsoft Access. VBA (Visual Basic for Applications) is used and gives students the opportunity to work with yet another implementation of programming concepts studied at the start of the course. Forms and macros from Access are studied with examination of the VBA code that is generated. This is followed by students constructing their own macros.

Once PHP and database concepts are covered, students work to create online databases with Web interfaces. This brings together a variety of the topics previously covered. The database management system that is used is MySQL, another open source product that has been designed to work seamlessly with PHP. Work in this application provides practice in algorithm design and improvement in programming skills.

Next, more work in scripting with VBA is applied to other tools from the widely used Microsoft Office suite. Customized macros are created for the spreadsheet Excel. Also, code is written to integrate documents created in Word, PowerPoint, Access, and Excel.

Following on the course menu is the ActionScript language provided with Macromedia Flash. A new scripting concept should be less confusing to students at this point based on their experience to date in scripting. This application enables students to create compelling animations using Flash and should be quite popular.

Java is used as the final programming topic for the course with an introduction to basic applet design. Java goes well beyond what can be considered as a scripting language. However, by this point in the course, students should have reached a level of sophistication which will allow them to at least construct simple applets in Java.

Recall that a major design goal of the course is to expose students to some of the elements of computer science in an exciting way, one which will hopefully motivate them to continue their studies in this discipline. Thus, in addition to the applications already mentioned, a variety of topics will be interspersed into course discussions throughout the course: ethics of technology, nanotechnology, artificial intelligence, wireless computing, computer security, cryptography, online communities, e-commerce, and peer-to-peer communication (to list some from a long list of available and interesting topics). Many of these topics are ones that were in the course design on the first iteration of the project.

During the early stages of the course, students will select a significant term project of their own choosing from a list of suggested topics. The project is to be a reasonably complex application that involves at least one type of scripting. A presentation at the end of the course is also required.

We believe that it is logical to conclude that by the end of this new course students are better prepared for the entry-level CS1 type computer science course where they can start serious "serious" algorithmic programming and problem-solving. Moreover, the course provides an interesting and non-intimidating environment for students who have the knowledge of a computer literacy course but who want to learn more about what is possible in computer science. And, a goal is to encourage ones in this last group to consider doing more work in computer science.

Furman University has done some preliminary testing of topics for this new course in the existing literacy/fluency course CS 16. In spring 2003, additional scripting modules were incorporated for testing and the term project focused on scripting. Student reactions were very positive. The new course is being offered in the spring 2004 semester at Furman (CS 17: Introduction to Computing with Scripting), team taught by Dr. Abernethy and Dr. Treu. Millsaps College plans to test some of the course concepts in the near future in its literacy/fluency course CS 1000.

Detailed information about course content is discussed in a paper, "Bridging the Gap Between Computer Literacy and Computer Science" presented in August 2003 at the Annual Conference of the LTSN Centre for Information and Computer Sciences in Galway, Ireland and can be found at

www.ics.ltsn.ac.uk/pub/conf2003/kevin_treu.htm

Material specific for the implementation of the course CS17 being taught at Furman in the spring 2004 can be found at

cs.furman.edu/~ktreu/cs17/

The authors have been involved in the information fluency activities of the Associated Colleges of the South from the planning and initiation of the program. We share the enthusiasm of many of our colleagues in ACS who understand that information fluency is vital for the liberal arts graduate in today's world. And, we believe that the final course developed in the project and described in this report can be a valuable tool in the student's path to information fluency.

Finally, the authors wish to express gratitude for support they received for this project from the Information Fluency Initiative of the Associated Colleges of the South funded through a grant from the Andrew W. Mellon Foundation.

March 2004