

LARGE PROGRAMMING PROJECTS FOR THE BEGINNING PROGRAMMER

Jeanne M Baugh Robert Morris University, baugh@rmu.edu
Paul J. Kovacs, Robert Morris University, kovacs@rmu.edu

ABSTRACT

This paper outlines a model for the teaching of a project-based introductory undergraduate course in computer programming. The students in this course had little to no background in programming. Strategies to allow the students to succeed with a large project in such a course are discussed. This model introduces the programming concepts at critical times during the duration of the project. The systematic process of creating a well-designed project leads to a significant coding effort on the part of the first time programmer. Beginning programming was taught to 546 students over a six and a half year period in this study. The primary assessment tool for the course was the successful completion of the semester programming project. The outcomes from the study indicate that this approach can be successful.

Keywords Computer Programming Courses, Project based learning, Novice Programmers, Computer and Information Systems Curricula

INTRODUCTION

The skill set required for employment in the “real world” is constantly evolving. [6]. Coding efforts are often large projects worked on by many individuals and comprising many, many lines of code. Employers want their employees to program as well as communicate and document their work. Programming skills along with communication skills go hand in hand for successful employment. In fact, employers are increasing demanding this of their entry level employees [8].

But how does one give the very beginning programming student the experience of writing and documenting a large program, when they don’t even have any idea of the concept of a variable. There are many building blocks of logic and syntax that must be mastered before a large program can even be attempted. Among these are identifiers, control statements, loops, sub-programs and class structures.

There are many different approaches to teaching programming. This can be evidenced in the number of programming text books available to the instructor. Two Instructors teaching the same course at the same University may be offering a completely different version of the same course material. Each Instructor has their own methods of teaching and the academic freedom enjoyed in the class room is something that is not taken lightly. [9].

According to The Accreditation Board for Engineering and Technology (ABET) [1], for an Information Systems Degree, students must take two semesters of a programming language. Often, the student will be required to complete a large programming effort in the second course, but such a project is also achievable in an introductory course, if it is structured well. There is research to suggest that requiring a large programming project in a first course can be successful. [5]. Because there are many basic topics a beginning programming student may know, it is often difficult to require a large project in the first programming course. But it is possible if programming topics are taught in a specific sequence.

This paper describes a method used to teach such a beginning programming course. This course can be an Introduction to Visual Basic, C#, C++ or Java. Specifically, the courses referred to in this research are C++ and Java. Because the topics in C++ and Java are very similar in a beginning course, this paper combines these two groups as one. Actually, as will be defined in the course structure, both sets of students were given almost exactly the same assignments each semester. This made it much easier for the authors to keep track of all work turned in by the students as the semester progressed. The students were also enrolled in both online and on ground sections and

most of the students came into the course with no programming background. Additionally, there was no significant difference in the ages between the C++ and Java students. Finally, an attempt was also made to combine the major principles of systems analysis and design with the tools, techniques, and procedures necessary to teach these beginning programmers.

The period of time for this research was twelve terms or four years, including four summer terms. There were 364 students enrolled in both the online and on-ground sections. The average age of the Information Systems students at the author's institution is 25. Class size ranges from 6 to 30 with an average size of 12. Of the 364 students in this study, about half are online and the others were in a traditional classroom setting. The Java and C++ courses referred to in this research were all taught by one of the authors at a private ABET accredited University in the Pittsburgh, Pennsylvania area.

COURSE PREREQUISITS

The prerequisites for these courses and the course descriptions are as follows:

INFS1020 - Intro Decision Support Systems - Provides the student with an understanding of how decision support systems using the capabilities of the computer can be combined with human intelligence to make better decisions. The course focuses on how information is collected, stored, organized, and managed to assist persons at all organizational levels make more effective decisions. Problem solving and decision-making are examined within the framework of information systems that use databases and spreadsheets as tools that assist with the processes. Specifically, students use Microsoft Excel to address such topics as "what-if" scenarios, sensitivity analysis, plotting and graphic displays. Microsoft Access is also used to gain experience in designing and creating effective databases, developing effective data interfaces, creating queries, and generating meaningful business reports.

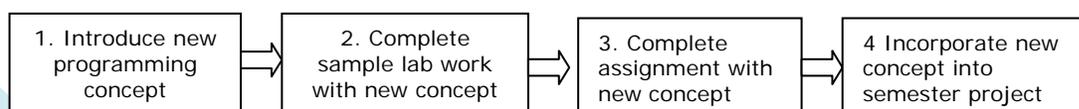
MATH2070 - Calculus W/Analytic Geom I - This is the first in a three-course calculus sequence. Topics covered include limits, continuity, derivatives, rules for derivation, applications, and related rates; optimization techniques for extremes including Rolle's and mean value theorems; first and second derivative tests; curve sketching; differentials and indefinite integrals; Riemann Sums; integration techniques, and the Fundamental Theorem of Calculus. The TI-83 (or higher) graphing calculator is required.

COURSE STRUCTURE

Researchers have investigated project-based learning in a wide variety of disciplines and settings and have generally found it to be effective in increasing student motivation and improving student problem solving and higher order thinking skills, addressing different learning styles, and providing students with an integrated learning situation [10, 2, 4, 13]. Project-based learning, unlike the traditional textbook/lecture approach, motivate the students to do additional work, illustrate to the students the value of the material covered, and most importantly, provide practical experiences that enrich the student's academic experiences.

Because the eventual outcome of the course is a significant programming project (approximately 1000 lines of code) it was essential to have a very structured approach to the course work. This structured approach applied not only to what had to be taught, but when it was taught.

The graph below indicates the basic progression of the programming topics in this course that eventually lead to a large programming effort.



Many Introductory programming texts teach the concept of classes at a very early stage in the coursework. However, it is the opinion of the authors that this concept is one that should be delayed until a firm understanding of basic programming concepts is fully understood, especially if the goal is the one defined here.

It has always been the practice of the authors to provide a great deal of sample code to the students, no matter what programming language is being taught. Short sample programs are an excellent way to illustrate a specific programming concept. Sample code is an essential part of both the online and on-ground Java and C++ courses described in this paper. As previous research as shown, this had proven to be extremely helpful to programming students [3].

Basic Assignments

The authors believe that first time programmers should not be concerned about how to run a program through either the C++ compiler or the Java compiler. Therefore, the students were always given a document containing step by step instructions, including screen shots of how to perform the tasks of starting the C++ or Java program, where to type in the code, how to compile the code and finally how to run the code. In fact, the first day of the on ground class always centered on typing in a very short sample program and compiling and running the code. The online students were also given very detailed instructions in the completing of this exercise.

The successful completion of each subsequent assignment also included the analysis and design of the problem, data requirements and logic needed to code and test the program. The students were often also required to turn in their pseudo code with each programming effort. McCauley viewed programming as a process of building a plan, in the form of source code, to achieve a certain goal [12]. Kazemian and Howles found that among the students they surveyed, only 5 percent of the students always developed a design prior to starting to program [11].

According to Wang, programming requires thinking with abstract concepts, which is difficult for novices. Second, programming includes many different tasks, such as problem solving, algorithm and data structure design, programming language comprehension, testing, and debugging. [14].

A number of small programming assignments (usually 4 to 5) were then designed to allow the student to master some of the basic building blocks of the language. A summary of the major topics covered within each assignment is as follows:

Assignment #1:

- Variable and memory allocation
- Input and output statements
- Simple assignment statements

Assignment #2:

- More complex math expressions
- If statements

Assignment #3:

- For loops
- While loops
- More complex conditional statements

Assignment #4:

- Beginning methods (Java) or functions (C++)

The on-ground students were given lab time each week to practice the concepts as they were presented. The online students were given these same exercises along with the solutions.

At about week four of a sixteen week semester, after the completion of the previous assignments, the students began work on the large programming project. A sample of an entire project is included in the appendix. Each semester the project is changed and thus tasks required in each phase were also changed. However, the example here is a good indication of the complexity level required of the students. Often the project is a real world case study of an organization, such as a hospital surgery department, a YMCA membership system or a College Registration System.

A sample summary of each phase and the required topics for that phase are as follows:

- Phase I - Completion of the entire structure of the menu driven programming project using stub methods/functions. Tasks to be mastered: Basic understanding of methods/functions is required.
- Phase II – Reading of data from various files into parallel arrays and reports generated from the data. Tasks to be mastered : Array definition, parallel arrays, array of a structure, reading and writing with data files, more advanced method/function topics
- Phase III– Modifying the array data and writing the data back to the data files, as well as changing the program into an implementation of a class. Tasks to be mastered: advanced input and output, and class implementation.
- Phase IV assigned– Additional reporting from the array data. Tasks to be mastered: advanced array manipulation, more on class definitions and the documentation procedures.

Systems Analysis Element

Each programming assignment also followed a systems analysis approach.

1. Analyze Requirements – What does the program have to do? This needs assessment closely follows the Analysis step in the traditional systems development life cycle (SDLC) and the instructor role is that of a project leader.
2. Design/Develop -Define the logic and various data needed to complete the project. The actual development of the program should be very structured and should follow the student's solution design. A deadline was set for the completion of each assignment and project phase.
3. Test/Document - It was stressed to these new programming students that testing of program inputs and output is essential. Kazemian and Howles found that only one-third of the students always static checked their work [11]. Documentation in the form of comment statements as well as the project documentation is also an important part of the course. Students were required to write both User's and Programmer's Guides.

COURSE RESULTS

As previously indicated, the authors have been teaching beginning programming courses for a number of years. The data used in this research encompasses the past 6.5 years, with a total of 546 students.

Those students who were able to complete most of the functionality of the programming project along with the documentation were given an "A" for the course grade. Even though a grade is assigned, what is really being assessed is the level of the student programs at the end of the course. Therefore, the actual exams administered during the term were weighted at a much lower value.

Because the project was used as the assessment tool, each semester a critical look was taken of the project. For example, reading data from files was moved to an earlier phase of the project because it was noted that the students seems to understand the concept at an earlier stage. Error checking for various data input was also added as each semester progressed.

Table 1 indicates the final grade for all of the students. It must be noted that there was no significant difference between the online grades and the on-ground grades. A great deal of help throughout the semester was given to these

programming students. As a result, the fact that so many students received an A is not surprising to the either author.

Help was provided to the students both in the traditional face to face format and online. The students would send their code as an attachment to their email along with questions. If the problem was small, the code would be fixed and sent back with comments placed in the corrected section of the program. If the problems were large, the student was asked to come in for a face to face meeting. If the student was not able to attend a face-to face meeting such as those taking the on-line course, a phone call would be made to the student and they would be talked through the problem.

Table 1. Student final grading

| Grade | Percent |
|----------------------|---------|
| A | 60% |
| B | 16% |
| C | 3% |
| D | 2% |
| F | 11% |
| Withdrew from course | 8% |

CONCLUSIONS AND RECOMMENDATIONS

There are number of approaches in the teaching of a beginning programming course. The method described in this paper has been extremely successful, considering that most of the students were first time programmers. The fact that 76% of the students received an A or a B in the course is directed related to the fact that a great deal of time was spent in helping the students de-bug their code.

Students were also very proud of their results. Watching the transformation of the students from not understanding a simple output statement to writing a large project encompassing many methods/functions is quite rewarding to the author.

Support for faculty is one issue that has not been addressed in this paper and must not be ignored. Although it is not a conclusion, it is a recommendation that Institutions must provide support for those Instructors who are willing to spend the additional time needed to bring very beginning programming students to this point of success. This support could be in the form of controlling class size and limited course preps. If the support is not there, the success of any such course cannot be assured [7].

Those teaching a course such as this must be very organized and attentive to the students. A policy of answering the student's email within 24 hours is extremely important. Students need to feel that the instructor is accessible. A study done by Yang and Cornelius also supports the concept that the students must receive feedback on a timely basis. [15]. Help was available and given when needed. One student said; "Without a doubt this was one of the best classes I have ever taken. The teacher was readily available; she responded to my emails fast and was very nice."

Often the programming books are difficult for the student to read, in that a specific programming concept may be embedded within a complicated example. Therefore, the sample code proved to be a valuable resource to the students.

Cheating is an issue for both on-ground and on-line education. It is also impossible to prevent it completely. But having the students explain their code either on a traditional exam or on an on-line exam (returned within a very

short return period) could help in this area. If the student does not do their own programming, it will be impossible for them to thoroughly explain what they turned in as their own work.

Assessment of course work is an area that all educators must not ignore. Furthermore, since the assessment of this course is done in terms of the completion of the project, the course has worked. Some argue that student exams are very important. The authors believe that learning to program is the most important outcome of this course, not the memorization of terms and programming rules. There should be a continuous improvement process by the instructor of any course taught to make sure that the goals of the course are met.

More research also should be done to look further into what works and what does not work for a technical course such as this one. The authors are also experimenting with allowing the first time programming student to design their own projects and data within a specific defined structure.

Not all students coming into such a course with no programming background will be able to successfully complete a large coding project. But this research has shown that it can be done if the course work is taught in a very specific, structured order. This data suggests that it is an option that can work if the student is willing to put in the time and energy along with the Instructor's commitment.

REFERENCES

1. Accreditation Board for Engineering and Technology (ABET) <http://www.abet.org>
2. Albanese, M. & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. *Academic Medicine*, 68(1), 52-81.
3. Baugh (2009). Student Success with Java Online vs Java On-Ground. *Information Systems Education Journal*, 7 (74). <http://isedj.org/7/74/>. ISSN: 1545-679X.
4. Buck Institute (1999). Project based learning. [Online]. <http://www.bie.org/pbl> [8 May 2000].
5. Davis, T. A. (2007). Graphics-based learning in first-year computer science. *Computer Graphics Forum*, 26(4), 737-737. doi:10.1111/j.1467-8659.2007.01034.x
6. Gallivan, Michael; Truex, Duane; Kyasny, Lynette, 2004, "Changing patterns in IT skill sets 1988-2003: a content analysis of classified advertising", *ACM SIGMIS Database*, Volume35, Issue36.
7. Gopalakrishnan, A.,(2006) "Supporting Technology Integration in Adult Education: Critical Issues and Models", *Adult Basic Education: An Interdisciplinary Journal for Adult Literacy Educational Planning*, v16 n1 p39-56 Spr. 18 pp.
8. Gruba Paul, Al-Mahmood Reem, (2004). "Strategies for communication skills development", *ACE '04: Proceedings of the sixth conference on Australasian computing education - Volume 30*
9. Hamilton, N., (2007) "Faculty Autonomy and Obligation", *Academe*, v93 n1 p36-42 Jan-Feb. 7 pp.
10. Hutchings, P. and A. Wutzdorff, 1988, "Experimental learning across the curriculum: Assumptions and principals." *New Directions for teaching and Learning*, 35, 5-19
11. Kazemian, F., and-T. Howles, 2008. Teaching challenges: Testing and debugging skills for novice programmers. *Software Quality Professional* 11, no. 1:5-12.

Issues in Information Systems

Volume 13, Issue 1, pp. 85-93, 2012

12. McCauley, R., S. Fitzgerald, G. Lewandowski, L. Murphy, B. Simon, L. Thomas, and C. Zander. 2008. Debugging: A review of the literature from an educational perspective. *Computer Science Education* 18, no. 2:67-92.
13. Tretten, R. & P., Zachariou, 1995, "Learning about project based learning." Paper prepared for the Autodesk Foundation, San Rafael, CA.
14. Wang, X. (O. P. H. I. E. (2010). Teaching programming skills through learner-centered technical reviews for novice programmers. *Software Quality Professional*, 13(1), 22-28
15. Yang, Y, Copnelius, (2004) "Students' Perception towards the Quality of Online Education: A Qualitative Approach", Association for Educational Communications and Technology, 27th, Chicago, IL. October 19-23 17 pp.

APPENDIX

Sample Programming Project:

Phase I:

Program displays the menu below and has the following functionality:

Area YMCA

1. Add/Modify Member Information
2. Add/Modify Class Registrations
3. Report Section
4. Exit the System

Please Make your selection >

Write the complete method for the menu and stubs for all of the other methods.

The names and definitions of the methods are as follows:

- Menu – method that will print the above menu and will return the user's selection by value to the main program
- Modify_member– method to add, delete, and change the basic member information
- Modify_registration- method to add, delete, and change class registration information
- Report – method to print reports

Phase II:

Add code to the following methods:

- Start_program -This method will be the first thing to run when the program executes. This method will open a text file called members.dat and read member information into parallel arrays. The member information that will be read in is as follows:
 - Member Code (four digit integer code)
 - Member Type (string) single, couple or family
 - First Name (string)
 - Last Name (string)
 - Age (Integer)
 - Gender (char - M/F)

- Report -When this method is called, the following menu will print and code required for the reports will execute:

Area YMCA

1. All Member Info
2. All Members of a Specific type
3. All Members between a specific age range
4. All Members of a specific gender
5. Report 5
6. Report 6
7. Report 7
8. Report 8
9. Report 9
10. Exit Report Menu

Phase III

Add code to the following methods to perform the functionality required. Also define a new class with all methods and arrays as members of this class.

- Add code to the Modify_member method to
 1. Add Member
 2. Delete Member
 3. Modify Member Information
- Add code to the Exit_program method to write all data from the parallel arrays back to the data file.

Phase IV

- Add the following variable definitions to the class and add code to the Start_program method to read in the following data from the course data file

| | |
|----------------|---------|
| Class Code | Integer |
| Class Name | String |
| Beginning date | String |
- Add the following variable definitions to your class and add code to the Start_program method to read in the following data from the registration data file

| | |
|----------------------|---------------------------|
| Register Member code | Integer (the member code) |
| Register Class code | Integer (the class code) |
- Add code to the Modify Registration Method to add, delete, or modify a registration.
- Add code to the exit_program method the write the data back to the data files
- Add code to the Report method to run additional reports:
 - Report 5 – List of all classes
 - Report 6 – List of all registrations (must include the class name and the member name)
 - Report 7 – List of all registrations in a specific class (prompt for the class code) You must also print the name of the member
 - Report 8 – List of all classes a specific member is signed up for (prompt for the member code)

Documentation required:

The Programmer's Guide:

- A short explanation of the program – this MUST include the explanation of how the program uses the data files
- A data dictionary – explanation of major data elements in the program along with
 - Classes
 - Methods within class – name and brief explanation of tasks
 - Data files
 - major variables – where used and what they are used for
- Sample screen shot of each of the reports

User's Guide:

A "How to" guide with instructions in the execution of the program