Enhancing Student Creativity in Information Systems Education: The Active Learning Approach

Amita Goyal, Ph.D.

ABSTRACT: *In the Active Learning Approach, an experimental teaching approach to enhance student creativity and participation, students are given the freedom to create their own problem search space and are given control of and responsibility for their decisions and actions. The goal of this approach is to increase student excitement, involvement, and learning by simulating a real-world experience in which the power of problem definition and structure is shifted from the instructor to the student. This paper discusses the Active Learning Approach and its implementation in an introductory programming class of an information systems curricula. The resulting student performance and attitude indicate a high level of effectiveness and benefit. Student interest, comprehension, awareness, and pride are heightened.*

KEYWORDS: *IS Education, Active Learning, Creative Learning*

INTRODUCTION

Most classes in information systems curricula consist of lecture presentations of important concepts followed by written evaluations to ascertain the level of student comprehension of the specified material. Additionally, some instructors attempt to enhance learning and comprehension through hands-on exercises, which provide a venue for the application of presented concepts. The problem with this method, however, is that students have very little, if any, control of or even input into the selection or structuring of the learning process. The lecture material, the written evaluations, and application projects are chosen, detailed, and structured by the instructor. Using spoon-fed tools, students mechanically perform the tasks assigned to them. Thus, students typically become passive rather than active learners.

Following a series of such structured and controlled course experiences, students are expected to perform in the job market. However, most students entering the job market struggle through the difficult transition from the university environment which demands controlled, encapsulated thinking to the real-world environment which demands creative, unstructured thinking. This is because as employees, students are no longer provided the comfort of the structured atmosphere; rather, they are expected to take control of situations, set parameters, and output creative solutions. As employees, these individuals are required to take active roles rather than the passive roles to which they became accustomed.

The business community is struggling with a current lack of creativity in the workforce [12]; the academic community is struggling to devise new and more exciting methods for knowledge dissemination to prepare students for the workforce [2]. Both communities share the blame for a current lack of creativity in students and employees, and are aggressively searching for and pursuing solutions to teach the application of creative thinking to business problems [12]. The business [8, 13, 15] and academic [10, 14] communities are rendering considerable support to active learning solutions.

This paper introduces the Active Learning Approach to help smooth the otherwise stringent university-to-job market transition, from the role of student to employee. Although this work concentrates on the implementation of the Active Learning Approach in academia, the concepts developed may be extended to business situations. The Active Learning Approach encourages active and creative learning by shifting the responsibility from the educator to the student, promoting an aggressive interest in learning, with an emphasis on the generation of creative ideas and active participation. This paper details experimental implementations of the Active Learning Approach in an introductory C programming course at Virginia Commonwealth University (VCU) and discusses advantages and benefits realized during implementation. Conclusions are based on the instructor's observations, on students' comments both in class discussions and on student evaluations, and in the high quality of the projects produced.

CHOOSING THE "TEST" COURSE

Although active participation and hands-on application should be stressed in all information systems classes, a programming class cannot exist without repetitive hands-on application and experimentation. Students enrolled in a programming class cannot be expected to grasp and understand a concept without numerous experimentation exercises which require direct application of the concepts. Habitually, instructors clearly define problem scope and specifically outline output expectations for each of the assigned programming exercises.

In the Information Systems Department at VCU, students are required, usually in the beginning of their junior year, to take BUS 358 - Introduction to Structured Programming Using C. These students typically have very little, if any, programming experience. During the past year, the Active Learning Approach was implemented in three sections of BUS 358. Each section met for one hour and fifteen minutes on two days per week and con-

Table 1

Project	Skills Learned
1	develop an algorithm for a basic input/output example, flowcharting, pseudocode, use C, typing, saving and printing out source code and output to a C program, arithmetic assignment statements; C syntax concepts: printf, scanf if
2	use C conditional structures and loops; algorithm development and refinement; C syntax concepts: if, for, do while, while, switch
3	continue using C conditional structures and loops; C syntax concepts: if, for, do while, while, switch, arrays
4	integrate and apply concept of strings, files and pointers in C

sisted of approximately 25 students. BUS 358 was chosen as the course in which to test the new teaching technique for the following reasons:

1) programming classes, especially introductory classes, are inherently difficult for most students.

2) since programming assignments stringently define problem scope and output expectations, students easily fall into the pitfall of structured, limited thinking and passive learning; students simply meet the specified requirements without experimenting and exploring their own ideas.

3) for most programmers, the transition from the classroom to the workplace is often very difficult; this is because in the workplace, problems and requirements are not clearly defined and a structured environment is not provided by the employer.

IMPLEMENTATION DETAILS

Because of its emphasis on unstructured, creative, unordinary thinking, the Active Learning Approach radically deviates from traditional teaching methods. Therefore, it was implemented on a very small scale, in order to study its effects on student performance and attitude while minimizing any potentially harmful outcomes of implementation. The specific goal of the implementation was to increase student excitement, involvement, and learning by simulating a real-world experience and shifting the power of problem definition and problem search space from the instructor to the students.

A discussion was held during the first class meeting in which the students were informed that they were expected to be creative thinkers throughout the semester. As a required expression of their creative abilities, they would have to participate in a self-created, team project (Project 5), which would be presented to the class during the final two weeks of the semester. So they could begin work immediately, the students were assigned Project 5 on the first day of class. The students were assured they would be provided a strong knowledge base of programming concepts in class (see Table 1) and were expected to creatively build upon this foundation of knowledge in order to complete the team project.

Building the Foundation - First Four Assignments

Five programming assignments were assigned to the students. Each assignment was worth ten percent of the final course grade (see Table 2). The first four assignments adhered to the traditional approach, where the specifics of each assignment were stipulated by the instructor. Detailed project specification sheets meticulously described a particular problem and specified exact input and output requirements. Additionally, each assignment listed what C constructs must be used in developing the source code. The students were given two weeks to complete each of these four assignments. They did not have any input into the specifics of the assignments.

To encourage creative thinking, up to ten additional bonus points were available for each of the first four programming assignments. However, to ease the eventual transition from the traditional to the active learning mode, the students were not given any specific requirements for these bonus points. Rather, they had to expand their source code by creatively building upon the concepts they had learned and the foundation of knowledge that had been built for the project. The students were empowered to:

1) choose whether or not to even try for bonus points; this required some thought because the students were given no guidelines on how to earn the bonus points.

2) determine creative extensions of the assignment in order to earn bonus points.

3) promote self-learning in order to successfully translate their specific ideas into C source code.

4) determine and recommend how many bonus points should be awarded for each of their creatively generated and implemented ideas.

There was no guarantee that their ideas would in fact earn bonus points or that they would receive their recommended number of bonus points for each idea. The bonus points provided an intellectual challenge for the students while providing them experience in creative and active learning.

After the completion of the first four projects, five weeks of the semester remained. The students were prepared to use the next three weeks to finalize developing and coding of Project 5 and the final two weeks of the semester presenting their work.

Project 5 - Student "Created" Assignment

Project 5 was a self-developed team project which adhered to the Active Learning Approach in lieu of traditional teaching methods. The professor did not specify any specific requirements for Project 5; rather, the students were given the control. Only the minimal guidelines were set by the instructor - the date the project would be due, the programming language that must be used to complete the project, the criteria for grading, etc. The students were encouraged to work on Project 5 throughout the semester (meet with their team members, select a problem to be solved, divide the problem into modules, assign modules to team members, etc.), although no checkpoints were set by the instructor. The specific goals of Project 5 included affording students the opportunity to:

1) unleash their minds to freely create, develop, and present their ideas.

2) increase excitement and involvement by simulating a real-world experience.

3) recognize what they had learned in the class and how it was useful and applicable to a real-world situation.

4) learn material beyond the rigid scope of the class by extending the framework of knowledge established during the semester.

5) take pride in their work, viewing it not just as an assignment, but rather as a display of their creative abilities.

6) work within a team as they may be required to do in the real world.

7) benefit from the diversity (as compared to their classmates) of the final projects.

Previous research has shown that engaging students in a client/development team relationship [9] and simulating the real-world environment [1] has yielded successful results. Therefore, Project 5 simulated a client/developer relationship between the instructor and the students. The instructor acted as the client in a fictitious corporation. The students were asked to immerse themselves in the role of a software development team (the students were allowed to choose their own team mem-

bers; however, in order to keep projects manageable, team sizes were limited to 4 students) that had been hired by the client to develop a software solution to address a problem of the corporation. Since the goal of the project was to empower the students with flexibility and control, the students were allowed to define a corporate problem of their own choosing. They were cautioned that the problem must be large and complicated enough to justify the joint effort of a team of four programmers. Teams with projects solving simple problems would be penalized in grading. Students were permitted to choose their own project plan (to span the entire semester) and set their own benchmarks and deadlines of deliverables to the client. The students were told to interact with the client as they would in a real-world situation. They had to judge if, when, how often, and at what stage(s) they would meet with the client in order to update the client on their progress and to obtain the client's approval.

Table 2

Exercise	Percentage of Total Grade
Programming Assignments (5)	50
Quizzes (2)	10
Midterm Exam	20
Final Exam	20
Total	100

"Most undergraduates resist teamwork. Previous research has shown that upon nearing graduation, however, undergraduates begin to recognize the value of team effort and a team's synergistic exchange of ideas which breeds innovation and creativity."

In order to encourage creative thinking, the instructor did not provide the students with any project ideas. Some students were initially discouraged because they could not immediately decide on a project. However, after a few meetings with their team members, all of the students managed to creatively hone in on a specific project idea. A similar approach was taken when students encountered roadblocks during development. To maintain the integrity of the real-world simulation, the instructor did not provide any assistance to the students regarding the specifics of the project (e.g., how can the project be enhanced, which specific resources should be consulted, etc.). So that the learning experience would not be compromised, the instructor did provide assistance with coding errors and algorithm logic and development.

The students were advised to develop a detailed Project Specification, which copiously described the problem being addressed, included an exhaustive modularization of the problem into clean components, and specified an assignment of equal responsibilities or module(s) to each member of the team. The students were encouraged to have the Project Specification approved by the client before delving into the project. Neither the development nor approval of the Project Specification, however, were required. Although very few teams actually sought approval from the instructor of a formal Project Specification, all of the teams chose to meet with the instructor several times during the semester to verbally discuss their project ideas, complexity, and details.

Resources

In order to ensure a real-world environment, no restrictions were placed on the resources available to the students. Students could use any resources available to them, including other faculty, students, business professionals, etc. The students could also exhaust any reference manuals and text (in addition to their prescribed textbook). In order to maintain academic and professional integrity, students were required to write their own source code. Since most professional programmers use routines written by other programmers and made available via public data banks, students could do the same to enhance their projects, provided they included proper references in their programs. (The instructor did not volunteer this information to the teams; however, some students discovered this while searching for creative enhancements to add to their projects.)

Students were given access to the vast repositories of information and resources that are available via the Internet. They were provided a personal Email account and were also given the instructor's Email address. Students could Email specific questions and problems, or even their entire program, to their classmates or to the instructor. Since the instructor had Email access from home, students were assured daily responses, even outside of office hours. Electronic communication and file exchange were encouraged, but not required.

An electronic class newsgroup was established and all students were given instructions on posting to and retrieving information from the newsgroup. All posts to the newsgroup were accessible to the instructor and the remainder of the class. Anyone in the class could read the postings and could post a response. To encourage usage, lecture outlines and solutions to assignments, quizzes, and exams were posted to the newsgroup by the instructor. The instructor initially feared that the availability of the newsgroup would serve as a disincentive for the students to attend lectures. However, since only lecture outlines, and not the complete lecture notes were made available, class attendance did not suffer.

Finally, a computer, loaded with a C compiler, and equipped with a color projection panel was made available to all of the teams. This paraphernalia was to be used for the official presentation to the client and corporate members (the instructor and the class).

Teamwork

Most undergraduates resist teamwork. Previous research has shown that upon nearing graduation, however, undergraduates begin to recognize the value of team effort and a team's synergistic exchange of ideas which breeds innovation and creativity [5]. This is further evidenced in a recent study of 800 in-

formation systems, computer science, and electrical engineering students [7]. In this study, over 50% of the respondents chose a specific employer because the employees in the firm work in small teams and with a diversity of assignments.

Like most undergraduates, BUS 358 students initially resisted the idea of working in a team. Initial concerns were that some team members:

1) do not regularly attend team meetings.
2) bring down the quality of work in the team, hurting the team's grade.
3) do not always complete their portion of the project.

A class discussion was held regarding the real world, in which programmers often work in teams. Team members have to adjust to different personalities and draw on the strengths of each member in order to complete a project. The students were assured they would receive an individual, not a team grade. And, if necessary, a team could fire one or more of its members if the member(s) did not attend group meetings, meet established deadlines, etc. A team member could not be fired simply due to personality conflicts. Being fired from a team resulted in a failing grade for the student.

Grading

Students were rewarded for the paradoxical efforts of creatively expressing their individual ideas, and yet, participating in the joint venture of their team [8]. Since a proper assessment of a student's abilities requires "different views of work - a finished product, a verbal performance, and documentation of aspects of the work process" [6], students were required to:

1) submit a written group report, including the project source code (electronic and hard copy).
2) individually demonstrate their portion of the project and field detailed questions from the instructor and the classmates during a formal class presentation.
3) discuss and evaluate the project development and teamwork process, including a formal evaluation (a grade of A, B, C, D, or F) and description of the contribution from each team member. Each grade assigned had to be formally justified.

Grades were computed as a weighted average of these components, with the heaviest emphasis placed on the difficulty and coding of the project and on the class presentation. Grades were assigned on an individual basis, not on a team basis. Students were graded on:

1) their coded portion of the complete program (including level of difficulty).
2) their portion of the formal presentation.
3) the evaluations they received from the remaining team members.

Bonus points were also assigned on an individual, not a on a team basis. However, regardless of the number of enhancements, bonus points were limited to a maximum of ten points per student.

IMPLEMENTATION RESULTS

The Active Learning Approach was implemented in three different sections of BUS 358, consisting of approximately 25 students per section. The results from all three implementations were very positive and consistent among the sections.

Projects

While a few students opted to minimize out-of-class research efforts, most students explored a number of additional texts and reference manuals to search for interesting projects, to add dimensions to their project ideas, and to resolve specific coding and debugging problems encountered. Students who were interested in pursuing careers as programmers capitalized on this opportunity to explore specific areas of interest.

The students developed skills that were not taught during the first four programming assignments (see Table 1). For example, all of the projects included menu interfaces. Most of the projects also included color. A number of the projects included 3D graphics, animation, color, and sound. Specific projects included inventory tracking systems, payroll systems, banking systems, ATMs, computer games, etc. Computer games (students assumed they were employed by a company producing computer games), some of which will be made available as Shareware on the Internet, included complex implementations of Blackjack, Russian Roulette, Pinball, and Slot Machines.

Overall, the quality of most of the projects exceeded the instructor's expectations. Had the instructor devised a traditional assignment in place of Project 5, the level of difficulty would have been far below that demonstrated in the students' self-developed projects. For example, the instructor would have assigned a project requiring the use of arrays and pointers; the instructor would not have required 3D graphics or animation be incorporated into the programs of an introductory C class. As a result of their creative and active learning efforts, the students engaged in and learned from coding complex projects. Consequently, they earned grades in Project 5 that were significantly higher than those earned in the first four projects.

Bonus Points

Approximately 90% of the students in all three sections consistently attempted to earn bonus points throughout the semester. Even if they were not rewarded for all of their ideas, they generally found the notion of bonus points for creative thinking to be enticing. In fact, many students opted to sacrifice some of the project requirements for the opportunity to be creative and to earn the bonus points. This behavior illustrates that students prefer the opportunity to creatively express themselves over stringent course requirements.

Team Evaluations

Students were very honest and concise in evaluating their group members. A few groups clearly identified the one or more delinquent and non-contributing group members. The instructor found a high level of consistency among the grades and justifications assigned to individual students by their respective team members. One group even fired one member of the group.

Electronic Newsgroup

The newsgroup was particularly useful for students who missed a class. They often used the newsgroup to retrieve the instructor's lecture outlines. Also, if students misplaced their copy of the syllabus, an assignment, a returned quiz, or solutions, they accessed the newsgroup and obtained an electronic and/or printed copy of the misplaced material. However, the newsgroups were not actively used for team correspondence and file exchange. Students preferred to use private Email, telephone, or personal meetings rather than posting to the class newsgroup.

Student Attitudes

Based on the instructor's observations and on the comments made in the course evaluations, student attitudes changed from passive to extremely active. When presenting their projects to the class, it was obvious that the students took pride in their creations, beyond just a concern for a grade. Students were no longer content with just doing well or just trying for some bonus points; they were energized and proud of their work. They struggled to present the best possible product they could and enjoyed the struggle! This was clearly evident in the high quality of the projects produced (e.g., 3D, animated card games and slot machines). In undertaking such demanding projects, the students performed far better than in the first four programming assignments. This is largely because the projects became creative, personal creations of each student and each team, rather than strict guidelines mandated by the instructor and ad-

hered to by the students. As each group presented their results, the remainder of the class took a genuine interest in the diversity and specifics of the project, asking thought-provoking questions.

In the course evaluations, over 95% of the students consistently stated that they felt the team project was a meaningful learning experience and by far, the best learning experience in the course. They felt future offerings of the course must include the Active Learning team project. This team project was better than other team projects in which they had participated because this project allowed them to be in control and to be the creators. In retrospect, students observed that they learned about the different areas and facets of programming, not just about a few specific programming constructs that were presented in class. For example, students commented that instead of learning just about arrays and pointers, they learned about using arrays and pointers to program graphics and sounds. Additionally, they learned about the facets of working in a team — both the advantages of modularization of tasks (drawing on the expertise and interest of each team member), and also about the difficulties of coordination.

Time Requirements from the Instructor

Although Project 5 required significant more planning on the part of the instructor (e.g., creating class time for project presentations, creating team evaluation forms, etc.), Project 5 was not particularly strenuous for the instructor for the following reasons:

1) since this was a programming course, the instructor's office hours were used extensively by the students during the entire semester for all aspects of the course, not just for Project 5. In fact, the instructor was somewhat less burdened with Project 5, because students met with the instructor in teams, rather than one at a time. Meeting with the teams did not require more time than a traditional Project 5 would have required.

2) the burden of grading was only slightly more than that of the first four projects. Most of the probing and testing of the program as well as the assessment of difficulty and quality of the project was performed during the presentations. Student evaluations of their team members had to be summarized, but spreadsheets made the task relatively untaxing.

SUMMARY AND CONCLUSIONS

Faculty struggle to develop innovative methods of knowledge dissemination; students struggle to learn concepts so they can apply them in their future jobs; businesses are in search of employees who are creative, strategic learners [3]. The combined strengths of classroom and experiential education can create an effective learning environment [4]. Although all classes in information systems curricula are challenging for both faculty and students, programming classes are particularly difficult. Students enrolled in programming classes often fall prey to passive rather than active learning. This paper has presented the Active Learning Approach, an innovative method to promote creative learning in the classroom. This teaching technique empowers the student to influence the learning process. In the experimental implementations of the Active Learning Approach in the introductory C programming class at VCU, students were given a basic foundation of programming concepts through four detailed and structured programming exercises; students were then given creative freedom to express themselves in a final, self-defined, team project.

Based on the high quality of the final projects as well as on the comments in the student evaluations, the Active Learning Approach was very well received by the students. The students enjoyed having creative freedom and felt it enhanced their learning process. The instructor was very pleasantly surprised at the excellent quality of the projects and in the positive attitudes of the students. Such positive results from empowering students with creative freedom have also been realized in other experiential studies. For example, students have been directly involved in the development and plan of an entire course [14] and have been provided hands-on experience in developing programming ideas and producing programs in a structured and monitored environment [11].

In becoming creative rather than passive learners, students learned to take charge of their learning process. This is particularly important for students studying technology for they cannot afford to be passive learners; they must actively pursue a course of self-directed learning in order to remain technologically current [13, 15]. The Active Learning Approach can easily be extended for other courses in information systems including databases and expert systems.

REFERENCES

[1] Bredemier, M.E. & Greenblat, C. S. (1981). The Educational Effectiveness of Simulation Games: A Synthesis of Findings. *Simulation and Games*, 12(3), 307-331.

[2] Burkhalter, B. B. & Schaer, B. B. (1984-1985). The Effect of Cognitive Style and Cognitive Learning in a Nontraditional Educational Setting. *Educational Research Quarterly*, 9(4), 12-18.

[3] Crane, K. From the Back Room to the Boardroom. (1989, April). *Computerworld*, 83-84.

[4] Dustin, D. L. (1981). Classroom and Experiential Education. *Improving College and University Teaching*, 29(4), 166-168.

[5] Harris, P. R. (1981, May). Synergistic Training and Education. *Training & Development Journal*, 48-71.

[6] Hawkins, J, Frederiksen, J., Collins, A., Bennett, D., & Collins, E. (1993, May). Assessment and technology. *Communications of the ACM*, 74-76.

[7] Kay, S. (1990, November). Looking for Mr. Good Boss. *Computerworld*, 117, 122.

[8] LeBleu, R. E. & Sobkowiak, R. T. (1993). Resolving paradoxes of change : Achieving purposeful change in IS. *Information Systems Management*, 7-9.

[9] Morrell, J. S., Freeman, J. L., Serrano, F. & Mock, R. (1993). Hands-on experience for students in information systems design. *Journal of Applied Business Research*, 141- 145.

[10] Nicholson, R. (1983). Managing Emerging Businesses. *Managing*, 20-25.

[11] Schiller, S. & Hanks, W. (1984, March). Experiential Learning Increases Students' Job Market Prospects. *Instructional Innovator*, 19-21.

[12] Schlossberg, H. (1993, March). Now adults have somewhere to go to think creatively. *Marketing News*, 8-9.

[13] Stokes, S. L. (1989, March). A Do-It-Yourself Education : IS Professionals Can Manage Change with a Proactive Approach to Training. *Computerworld*, 117.

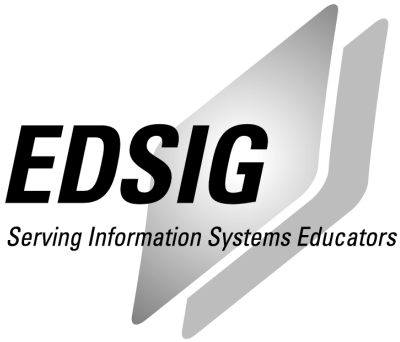
[14] Thomson, N. S. (1994). Using TQM Principles to Teach Current Topics in Information Systems. *Journal of Information Systems Education*, 65-71.

[15] Woodard, W. A. (1990, July). Learning to Cure Technical Obsolescence. *Datamation*, 75-76.

Amita Goyal, Ph.D.
Department of Information Systems
School of Business
Virginia Commonwealth University
1015 Floyd Avenue
Richmond, VA 23285-4000
Phone: (804) 828-7131; FAX: (804) 828-8884
Email: amita@cabell.vcu.edu

Dr. Amita Goyal is an Assistant Professor in the Information Systems Department at Virginia Commonwealth University. She received her B.S. in computer science and M.S. and Ph.D. in information systems, all from The University of Maryland at College Park. Her research interests include distributed database systems and multimedia and curricula integration. Earlier this year, Dr. Goyal served as Program Chair for EDSIG's Worldwide Conference on Information Systems Education (WISE 1995). Dr. Goyal is a member of the Association of NeXTSTEP Developers, Inc. (ANDI).

Information Systems & Computing
Academic Professionals



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096