Right From the Start: Leveling (then Raising) the Playing Field

Track: Women and Minorities in Computing

Thad Crews, Jeff Butterfield, and Ray Blankenship

Department of Management and Information Systems

Western Kentucky University

Bowling Green, KY USA

(270) 745-4643

Email: thad.crewsii@wku.edu, tjbutterfield@yahoo.com, ray.blankenship@wku.edu

Right From the Start: Leveling (then Raising) the Playing Field

Abstract

This paper covers an approach to teaching and learning programming that emphasizes logic and design while minimizing the distraction from hardware. This approach provides a gender-level playing field by removing many of the known problems associated with an introductory programming course. This paper also presents the Visual learning tool, which utilizes flowcharts to emphasize patterns of logic and design, abstracts out the details of different hardware, and supports a broad range of programming and problem solving activities. Visual is able to execute flowcharts, providing students with the immediate feedback that is typically only available with traditional high-level programming languages. We report the results of an empirical study that demonstrates and increase in performance for both female and male students, while also reducing the differences between males and females. This paper concludes with a discussion of implications and future directions of this research effort.

Keywords: Gender, information technology, introductory computer programming

INTRODUCTION

A report from the Presidential Information Technology Advisory Committee indicates a critical need for a diverse IT workforce if the United States is to meet the challenges of the information age (PITAC, 1999). It is a well-known and documented phenomenon that the Information Technology industry suffers from gender inequity, both at post-secondary institutions and in the workforce (Camp, 1990; Freeman, 1999; NSF, 2000). Just as there is no silver bullet for taming the complexity of large software projects (Brooks, 1986), there is likewise no magic elixir to cure the gender ills. The shrinking pipeline problem is complex and multi-faceted, and progress requires incremental advancements in many areas, including removing gender bias in computer software, increasing female access to and experience with computers, improving the awareness of female role models, and changing the perception of computing culture (Davies 2000; Thom, 2001; Woodfield 2000). It is not enough that we encourage capable females to enter the pipeline. We must also maintain a level playing field for those who do enter and avoid the "leaking pipeline" problem of losing women who are on track for success in Information Technology (Alper, 1993; Humphreys, 2002).

Rather than restating any of the abundance of available data regarding the problem, this paper proposes to offer one solution for improving the experience of female students in the pipeline at a key point in their IT education: the first programming course. The following section provides an overview of our approach and how it serves to provide a gender level playing field. We then present an empirical study showing how our approach raises the performance of students equally well for males and females alike. The paper concludes with a discussion of implications and future directions of this research effort.

RIGHT FROM THE START

Right From the Start is a five year ongoing research effort at (Anonymous) University to improve the learning experience in the introductory programming course. The objectives for our project begin with teaching programming by emphasizing the logic and design rather than the "power tools" of the industry. Second, we make efforts to decrease the distraction from hardware. Third, we require students to complete numerous small programs to develop a sense of repetition that will compensate for differences in individual backgrounds. Finally, we treat the programming language as a tool for problem solving. Specific languages come and go, but problem-solving skills are essential for success in the IT industry. While these have been the goals of our research effort for over five years, it is comforting to see other data and opinions that support our goals. Research has shown that girls dislike narrowly and technically focused programming classes (Countryman, 2002). Interested readers are also encouraged to consider (De Palma, 2001) whose suggestions are strikingly similar to our project goals, but not necessarily for the same reasons.

To accomplish these goals, we utilize the Visual Development Environment. Visual was developed in conjunction with the *Right From the Start* research project, and was designed to support our project objectives. Visual is easy for novice programmers to learn and use. Visual emphasizes patterns of logic and design, abstracts out the details of different hardware, and supports a broad range of programming and problem solving activities. Visual avoids the complexity of a high level programming language by utilizing flowcharts to represent logical solutions. Logical solutions are created by adding, deleting and moving flowchart elements using an intuitive point-and-click, drag-and-drop interface. Figure 1 shows the flowchart design window with a one possible solution to an overtime problem.

INSERT FIGURE 1 ABOUT HERE

The most significant contribution of Visual, however, is its ability to execute flowcharts, providing students the kind of immediate automated feedback previous only available with a traditional high-level language. When using the Visual tool, students may design, develop, execute, test and evaluate computer programs without any exposure to a traditional high-level language. Creating the "Hello World" program is just as easy as selecting an output flowchart element and entering the greeting. There is no need to explain libraries, I/O streams, or anything beyond the logic of the solution.

Likewise, the classic overtime problem is easily solved using flowchart elements as shown in Figure 1. When the student selects "Run" the Visual System executes the flowchart, including input dialog prompts and the appropriate data in an output window.

Visual was developed as part of the *Right From the Start* project for the purpose of providing novice students with a solid foundation for success when programming. Visual supports a broad range of activities (e.g., input, assignment, output, conditions, loops, variables, procedures, etc.) while minimizing the specifics of hardware and language syntax. We believe that female students will benefit from the Visual tool as it levels the gender playing field by removing many of the known problems associated with an introductory programming course, such as boys having an experience advantage with traditional "power" languages such as C++ and Java, and boys having more interest in the hardware that underlies the logical solution. At the same time, we anticipate male students will also benefit from Visual and the project goals mentioned earlier. The following section is a description of an empirical study showing how our approach raises the performance of students equally well for males and females alike. A discussion of implications and future directions of this research effort follows.

EMPIRICAL STUDY

The purpose of the reported study was to examine the effect of incorporating the Visual tool into a first semester programming course.

> *Hypothesis 1*: Female students in the treatment group will demonstrate a superior understanding of programming logic over female students in the control group.
>
> *Hypothesis 2*: All students in the treatment group will demonstrate a superior understanding of programming logic over all students in the control group.

Hypothesis 3: Differences between male and female students in the treatment group will be less than differences between male and females in the control group.

Method

Participants: The subjects for this study were 73 undergraduate students at a comprehensive public university with a traditional 16-week semester. The courses were 200-level introductory Visual Basic (VB) programming taught by two computer science professors with well above-average teaching evaluations. Students in the courses came from different backgrounds and had different majors. Both instructors employed similar textbooks and course content except for treatment. Subject assignment to class sections was semi-random as registration priority was done on the basis of last-name order.

Apparatus: The programming logic component of the experiment followed a design similar to (Scanlan 1989), involving three programs of varying difficulty, classified as simple, medium, and complex, shown in Figure 2. Answer sheets were constructed containing three data sets of initializing values and places for participants to write down the output for each data set, the time when they completed the three data sets, and the confidence ranking of their solutions.

INSERT FIGURE 2 ABOUT HERE

Procedure: Students assigned to control group were trained in VB using what (Brusilovsky, 1997) defines as a 'traditional' approach. This pedagogy used a VB manual for a text, a lecture format in class, and simple VB programming assignments as homework. Students assigned to treatment group spent the first five weeks of the semester learning problem-solving approaches and programming logic. A series of short programming activities using Visual were assigned as homework. No VB instruction was provided during this time. After the first five weeks, the treatment group sections began using the same pedagogy and materials as the control group sections. Note that the notion of spending between three to five weeks for a syntax-free introduction to programming is has also been suggested elsewhere (e.g., Schneider, 2001; Shackelford, 1999). The measurements occurred for both groups during the 16th week of the semester as part of a regular class meeting (see Figure 3).

INSERT FIGURE 3 ABOUT HERE

The measurement took place at the end of the semester. Participants were provided the simple program with a random answer sheet. Participants determined the output for each of the three inputs provided on the answer sheet. After determining the three output sets, the participants recorded the time as displayed on a projection screen. Finally, participants recorded their confidence level on a 5-point scale (1 = not confident, 5 = very confident). All participants were given sufficient time to complete the task. This process was repeated for the medium program and then the complex program.

Scoring: Individual correctness scores were determined by awarding one correctness point for each of the three outputs that were correct per solution sheet. Half credit was given to outputs that were correct through the initial loop iteration, but incorrect at some later point. The time data was recorded as the number of seconds required for determining the outputs. The confidence score was recorded from the 5-point scale. This grading process was the same for each of the six answer sheets provided by each participant.

Results

Hypothesis 1: According to Hypothesis 1, female students in the treatment group will demonstrate a superior understanding of programming logic over female students in the control group. In support of

Hypothesis 1, females in the treatment group scored significantly better than females in the control group on all three programs.

The simple program contained a nested-if statement and no loops. For this program, the treatment female group average score was 2.75 and the control female group average score was 2.10, which was a significant difference under ANOVA analysis ($p <= .01$). The medium program contained two nested-if statements inside a single loop. For this program, the treatment female group score was 2.13 and the control female group score was 1.15, a significant difference ($p <= .01$). The complex program contained two loops and four conditions. Once again, the treatment female group scored significantly better than the control female group (1.66 to 1.10, $p <= .01$). The results for all three programs are shown in Figure 4.

INSERT FIGURE 4 ABOUT HERE

Hypothesis 2: According to Hypothesis 2, all subjects in the treatment group will demonstrate a superior understanding of programming logic over all students in the control group. In support of Hypothesis 2, the treatment group scored significantly better than the control group on all three programs. The treatment group achieved scores that were significantly higher than those of the control group for all three programs, simple (1.74 to 1.10, $p <= 0.01$), medium (1.74 to 1.10, $p <= 0.01$) and complex (1.74 to 1.10, $p <= 0.01$). The results are shown in Figure 5.

INSERT FIGURE 5 ABOUT HERE

Hypothesis 3: According to Hypothesis 3, differences between male and female students in the treatment group will be less than differences between male and females in the control group. Again the hypothesis was supported by the data. An analysis of total scores for all three problems shows that both male and female students from the treatment group outperformed their counterparts in the control group. It is equally important to note that the data also shows that the difference between sexes was quite small for the treatment group (6.53 to 6.77, 3.5%). The difference between sexes in the control group is much greater (4.35 to 5.07, 14.2%). Both the improvements resulting from treatment and the reduction in gender differences can be seen in Figure 6.

INSERT FIGURE 6 ABOUT HERE

These findings suggests that providing a foundation for programming through a series of small logic problems using a pragmatic tool that minimizes hardware and language syntax details creates a gender level playing field, and that the level of that playing field rises for all students regardless of gender.

DISCUSSION AND IMPLICATIONS

The findings of the study are encouraging. The *Right From the Start* approach of using the Visual tool to teach foundational programming concepts in a hardware and language syntax free environment was well received. Student comments regarding the Visual tool was strongly positive by both males and females. When a traditional high-level programming language was introduced in the 6^{th} week, the female students had already established their ability and right to succeed in the course, both in their minds and in the minds of the male students. Traditional concerns for an introductory programming course (e.g., the experience gap with traditional programming languages between males and females based on self-study; the unnecessary bias towards hardware in a programming course; etc.) did not seem to arise. By leveling the playing field in this way, we are able to create an environment where women succeed at the same rate as their male counterparts. And succeed they all did! The *Right From the Start* project not only levels the playing field, but also raises the level playing field for all students, regardless of gender, age or nationality.

The problem of declining female participation in the computing disciplines is the subject of much speculation. Many of the proposed solutions involve activities outside of the control of the classroom, such as presenting young girls with gender-friendly computer games and increasing the number of tech-savvy teachers. These and many other excellent long-term improvements will likely prove beneficial over time. The *Right From the Start* project, however, is intended to have a measurable effect on students currently in the pipeline with the goal of reducing leaking by leveling the playing field and providing a foundation for future success. The contribution of the *Right From the Start* project is important in that it achieves its goals for a specific time in the development process of an IT professional. We hope that our success encourages other researchers, teachers and IT professionals to identify and develop methods for leveling and then raising the playing field for all individuals in the pipeline for a career in Information Technology. Each contribution—like each person—matters.

References

Alper, J. (1993). “The Pipeline is Leaking Women All the Way Along,” *Science,* 260, 16, pp. 409–411.

Brooks, F. (1986). “No silver bullet—essence and accidents of software engineering,” Information Processing 86, H. J. Kugler, Ed. Amsterdam: Elsevier Science (North Holland), pp. 1069-1076.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., and Miller, P. (1997). “Mini-languages: a way to learn programming principles,” *Education and Information Technologies,* 2, pp. 65–83.

Camp, T. (1990). “The Incredible Shrinking Pipeline,” *Communications of the ACM,* 33(11), pp. 47-57.

Countryman, J., Feldman, A., Kekelis, L., and Spertus, E. (2002). "Developing a Hardware and Programming Curriculum for Middle School Girls," SIGCSE Bulletin, Vol. 34, No. 2, pp. 44-47.

Cukier, W., and Shortt, D., and Devine, I. (2001). "Gender and Information Technology: Implications of Definitions," Proc. Information Systems Education Conference, Cincinnati.

Davies, A., Klawe, M., Mg, M., Nyhus, C., and Sullivan, H. (2000). "Gender Issues in Computer Science Education," Proc. National Inst. Science Education Forum, Detroit.

De Palma, P. (2001). "Why Women Avoid Computer Science," *Communications of the ACM 44*, 6, pp. 27–29.

Freeman, P. and Aspray, W. (1999). "The Supply of Information Technology Workers in the United States," Computing Research Association Report, http://www.cra.org/reports/wits/cra.wits.html (accessed June 13, 2002).

Gorriz C. and Medina C. (2000). "Engaging girls with computers through software games," *Communications of the ACM 43*, 1, pp. 42–49.

Humphreys, S. and Spertus, E. (2002). "Leveraging an Alternative Source of Computer Scientists: Reentry Programs," SIGCSE Bulletin, 34, 2, pp. 53-56.

NSF. (2000). "Women, Minorities, and Persons with Disabilities in Science and Engineering," National Science Foundation Report, http://www.nsf.gov/sbe/srs/nsf00327/start.htm (accessed June 13, 2002).

Presidential Information Technology Advisory Committee, (1999). *Information Technology Research: Investing in our Future*. < http://www.itrd.gov/ac/report/ > (accessed June 13, 2002).

Scanlan, D. (1989). "Structured Flowcharts Outperform Pseudocode: An Experimental Comparison," *IEEE Software*, pp. 28-36.

Schneider, M. and Gersting, J. (2000). *An Invitation to Computer Science, Java Version*, Brooks/Cole, Pacific Grove, CA.

Shackelford, R. (1998). Introduction to Computing and Algorithms. Addison Wesley.

Thom, M. (2001). *Balancing the Equation: Where Are Women and Girls in Science, Engineering, and Technology?* The National Council for Research on Women: New York, NY, 2001.

Woodfield, R. (2000). *Women, work and computing*. Cambridge University Press, UK.

Webb, N., and Shavelson, R., (1985). "Computers, education, and educational psychologists," *Educational Psychology, 20*, 4, pp. 163-165.

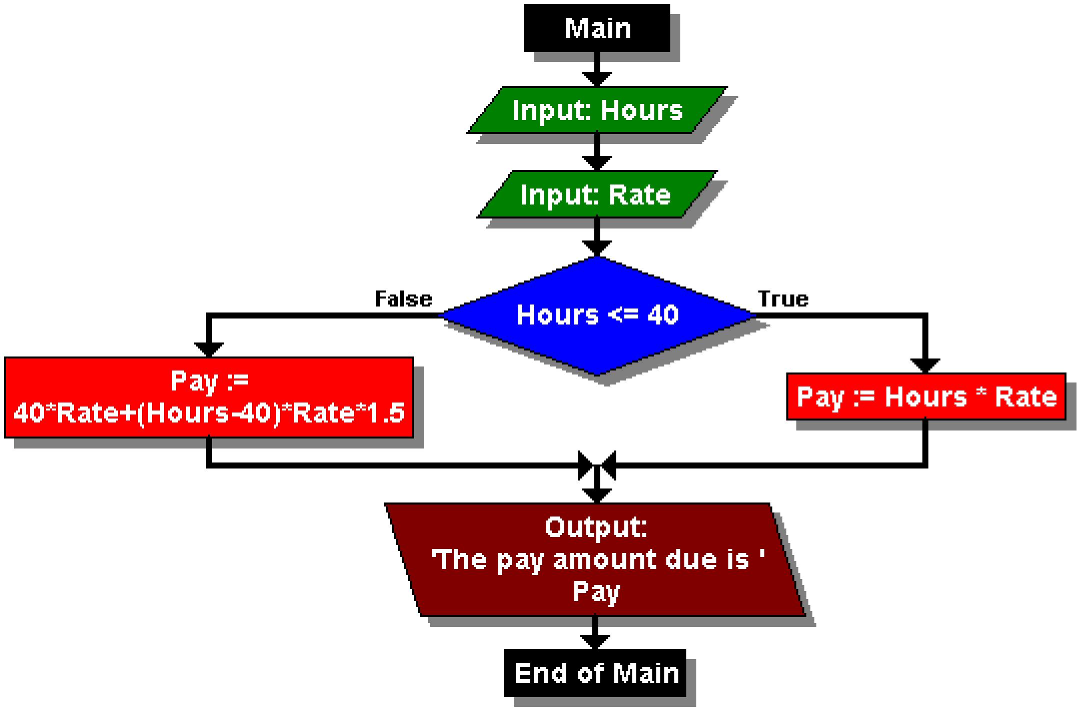

Figure 1: Visual solution to overtime problem

EASY PROBLEM

```
If A < D Then
   Print D
Else
   Print A
   If B < C Then
      Print C
   Else
      Print B
   End If
End If
```

MEDIUM PROBLEM

```
Do While A < 4
   If A < D Then
      Print B
   Else
      Print A
      If C < B Then
         If A < B Then
            Print C
         Else
            Print D
         End If
      Else
         Print A
      End If
   End If
   A = A + 1
Loop
Print C
```

COMPLEX PROBLEM

```
Do While B <= 6
   If D > B Then
      Do While C >= 2
         Print C
         C = C - 1
      Loop
      If A < B Then
         Print D
      End If
   Else
      Print B
      If B < C Then
         If A < B Then
            Print A
         Else
            Print B
         End If
         Print C
      Else
         Print D
         If A < B Then
            Print B
         End If
      End If
   End If
   B = B + 1
Loop
Print C
```

Figure 2: Simple, Medium, and Complex problems

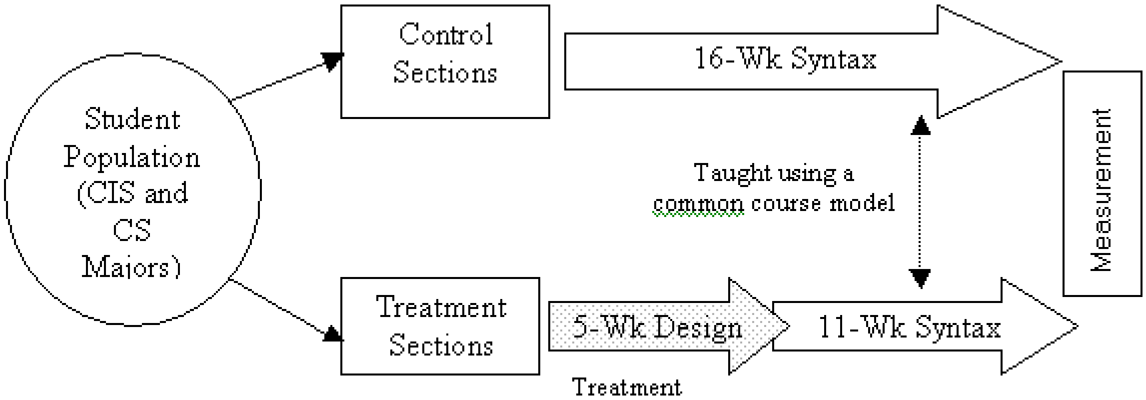

Figure 3: Research Process

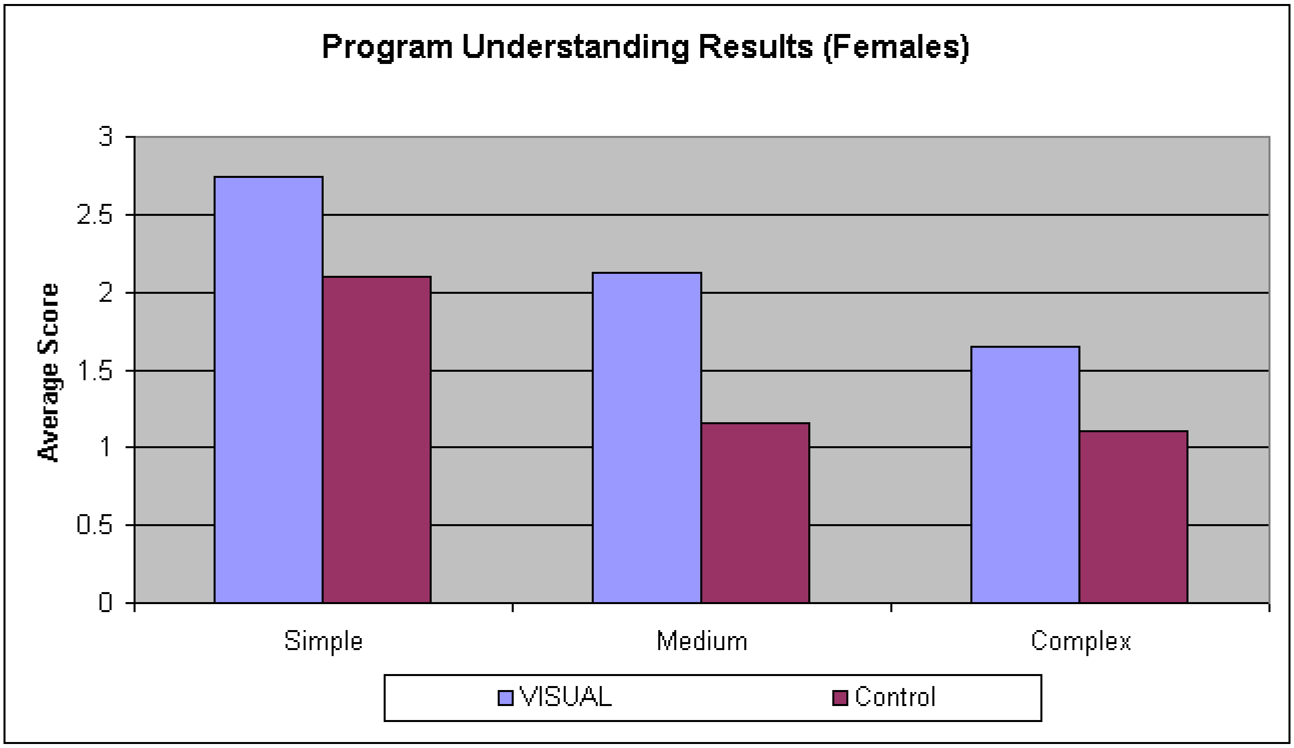

Figure 4: Hypothesis 1 Results (Females)

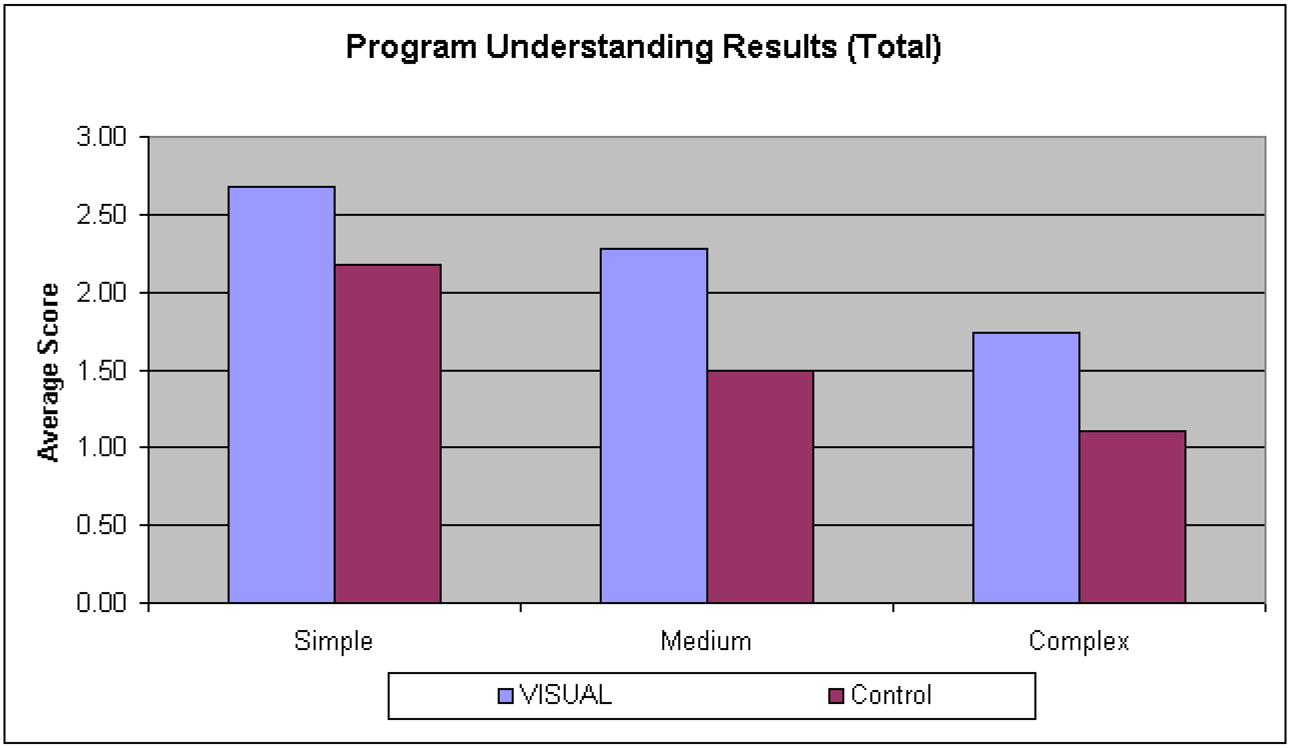

Figure 5: Hypothesis 2 Results (All Students)

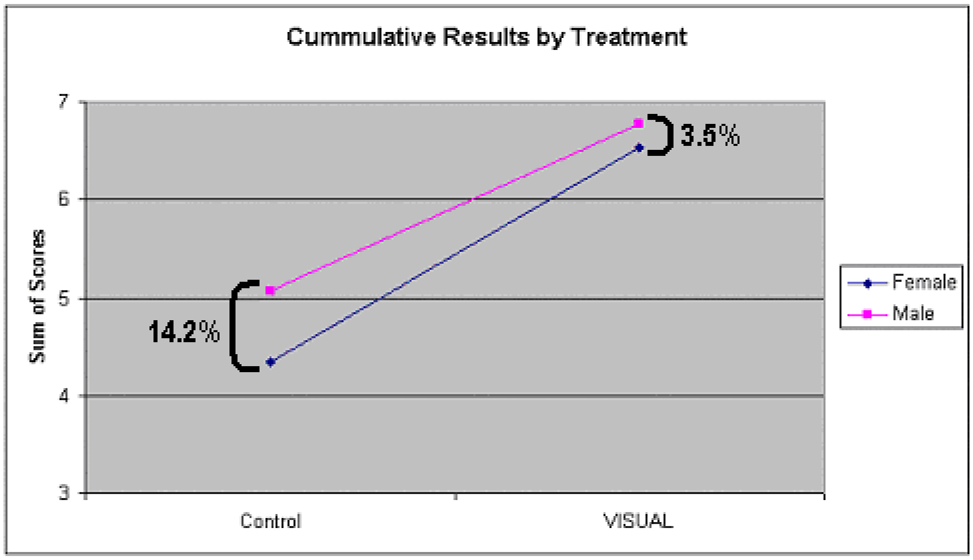

Figure 6: Hypothesis 3 Results (Gender Differences Between Groups)