# Analysis of Student Performance in Programming Subjects of an In-house Exit Exam

Masoud Naghedolfeizi<sup>1</sup>, Singli Garcia<sup>2</sup>, and Nabil Yousif<sup>3</sup>

**Abstract-**This paper analyzes the performance of students in Computer Science (CS) and Computer Information Systems (CIS) in the programming subjects (Principles of Programming I & II and Data Structures) of an in-house exit exam given in years 2003-2006. The student performance was measured in seventeen different categories of computer programming. The data for each category was colleted based on student responses to questions related to that particular category. The data was analyzed with respect to whether a student was majoring in CS or CIS, and with respect to whether a student was a male or female. The analysis of data showed that students majoring in CS generally performed better than CIS students in most categories of computer programming. Additionally, both male and female students performed approximately at the same level. The results of the assessment analysis will be used to identify the problem areas and make necessary adjustments to both curriculum and teaching strategies in order to improve and enhance the long-term knowledge retention of students in computer programming subjects.

#### Introduction

Fort Valley State University is a land-grant historically black institution with a strong commitment to bring high quality education and public service programs to its students and community. The University is located at Middle Georgia and serves over 2000 students through its comprehensive undergraduate and graduate programs.

The Department of Mathematics and Computer Science has a yearly enrollment of approximately 120 students in CS and CIS majors. The department offers two on-campus degree programs: a B.S. in Computer Science and a B.S. in Computer Information Systems. Currently, there are five full-time faculty members and one part-time member who are teaching program courses of both majors.

The curricula of both majors require students to complete at least ten (10) semester hours of common programming courses, which include seven (7) hours of programming I and II and three (3) hours of data structures. Computer science students must also take C/UNIX and Analysis of algorithms courses whose topics are mostly programming related materials. Students majoring in computer information systems must also take two additional programming courses; namely, COBOL and File Processing. Both majors may also take three to six hours of junior level elective courses in programming.

The outcomes assessments for computer programming courses are carried out in two parts. In the first part, a Sophomore diagnostic project is given to students in the Programming II in order to evaluate student ability to apply and integrate various programming topics to design a program for a relatively complex problem. In the second part, a separate assessment exam is utilized to measure specific knowledge level and programming skills of students. The exam is carried out in the data structures course. The assessment results are reported to the department, college and university for program review, teaching strategies, and accreditation purposes. The aforementioned programming courses will be also used in a yearly departmental exit exam.

<sup>&</sup>lt;sup>1</sup> Fort Valley State University, 1005 State University Dr., Fort Valley, GA 31030, feizim@fvsu.edu

<sup>&</sup>lt;sup>2</sup> Fort Valley State University, 1005 State University Dr., Fort Valley, GA 31030, garcias@fvsu.edu

<sup>&</sup>lt;sup>3</sup> Fort Valley State University, 1005 State University Dr., Fort Valley, GA 31030, yousifn@fvsu.edu

This paper analyzes the performance of students in Computer Science (CS) and Computer Information Systems (CIS) in the programming subjects (Programming I & II and Data Structures) of an in-house exit exam given in years 2003-2006.

#### **Outcomes Assessment**

In recent years, there has been an increasing trend in undergraduate computer science programs to assess programming knowledge and skills of students after completing the first introductory programming classes. The advantages of outcomes assessment in programming courses are discussed and presented in various pertinent publications [1, 2]. From an educational standpoint, one of the most useful benefits is that assessment provides a standard by which student learning and teaching strategies are evaluated both internally and externally (e.g.; by accreditation organizations). For instance, a careful analysis performed in reference [1] suggests that assessment data were central in identifying problem areas that resulted in poor student performance. Based on their findings, the problems were associated with bad programming habits of students, teaching object oriented programming in the first programming course, and lack of sufficient lab time. Similar findings and analyses have also been reported by other researchers and educators [3-5]. Additionally, methods of assessment and suggestions for future works have been discussed in the literature to address issues related to students with different programming and mathematical background and the types of questions used in the assessment [6-8].

While the value of outcomes assessment in programming has been noted in many recent publications, the focus of analysis has been mainly based on short-term assessment results obtained during or right after a programming course. A long-term assessment of programming courses could reveal additional problems such as the student ability to retain programming knowledge and demonstrate satisfactory programming skills. A long-term assessment could be carried out in a comprehensive exit exam during the student's senior year.

### **Exit Exam**

The curricula of CS and CIS programs at FVSU require students to take an exit exam as partial fulfillment of their degree programs. Prior to 2003 academic year, students were required to take the Graduate Record Exam (GRE) as the exit exam. However, the departmental program reviews, performed from 2000 to 2002, revealed that a new comprehensive assessment method would be needed for a more systematic and accurate evaluation of knowledge and skills of graduating students as required by the university new outcomes assessment policies. Thus, in spring 2003, the department decided to use an in-house exit exam instead of the GRE.

The exit exam includes topics from both core and major courses in computer science. The common subjects of CS and CIS exit exam include Programming, Database, and Data Communications. Each major also has its own specialized subjects. The specialized subjects for CS include: Digital Fundamentals, Operating Systems, Analysis of Algorithms, and Computer Organization. For CIS the subjects include: Information Technology, Information Theory, System Design and Analysis I and II, and COBOL. The common subjects are given in multiple choice format and specialized topics in essay format.

The new exit exam is a better and more reliable tool than GRE to identify strengths and weaknesses of graduating students in knowledge retention of broad areas within computer science and information systems. In addition to its importance to CS and CIS curricula, the exit exam would also be more inline with Association of Computing Machinery (ACM) accreditation requirements.

The first exit exam was given in Spring of 2003 with a flexible passing grade requirement in order to establish baseline data for a required passing grade. In subsequent years, the department set a passing score at 70% mark or better. Students who could not pass the test in the first try would be given a second chance to take the test two weeks after the first test.

The computer programming categories in the exit exam are selected from the following core courses: Principles of Programming I & II and Data Structures [9-11]. The exam questions are designed to cover major categories of

programming at the fundamental knowledge level expected from all graduates in the CS and CIS majors. Table 1 shows the programming categories and content of a typical question for each category.

	T		T-mi-sl		T
Categories	I ypical	Categories	I ypical	Categories	I ypical
	Question		Question		Question
	contents		contents		contents
Basic	Input output	Functions	Parameter	Stacks	Formation and
computer	device, CPU		passing and		operation
Knowledge	operation		type		
			identification		
Simple Data	Identification	Computer	Binary	Queues	Formation and
Types	of integer, float	Algebra	arithmetic		Operation
	and character	-			_
Syntax	Basic C++	Arrays	Arithmetic	Pointers	Definition,
	programming	-	operation,		identification,
	syntax		searches		and
					application
Selection	If and case	Records	Identification	Linked Lists	Type
structures			and		identification
			assignments		and traversing
Repetition	For and while	Recursion	accumulation	Trees	Binary tree,
structures	loops and				searches
	accumulation				
Problem	Identification	Classes	Objects and		
solving	of the best		properties		
_	solution		. –		

Table 1: Programming Categories used in the Programming Subjects of exit exam

# **Data Collection**

We have collected student performance data on the programming subjects of the exit exam for the last four years. The data has been organized according to the programming categories shown in Table 1. In addition, the data was arranged based on student major and gender. The purpose not only is knowledge assessment, but also is to study the possible role that major and gender play in the student performance. Table 2 shows the data collection performed according to the above description. The data in the table is reported in percentage of correct answers. For example, if a category has 4 questions and out of 20 students 10 answered all correctly, 5 answered 3, 3 answered 2, 2 answered 1 and 1 answered none, then percentage of correct answers is computed as follows [(10\*4+5\*3+3\*2+2\*1+1\*0)/(20\*4)]% = 79%

Table 2: Selected Programming Categories Used in Exit Exams during 2003-2006

		2	CDT	2			C4.7		670	0	TD			Dia	<b>CN4</b> A	
		Syn	201	22	FNI	ARI	Str	CL3	310	Que	IR	LNK	RCO	Prs		OVL
2	CS	72.5	80.0	80.0	60.0	63.3	60.0	37.5	100.0	100.0	80.0	80.0	80.0	55.0	63.3	64.5
0	CIS	63.5	34.6	53.9	50.0	30.8	51.3	34.6	61.5	61.5	69.9	76.9	38.5	50.0	43.6	48.5
0	М	75.0	67.0	100.0	50.0	55.6	77.8	33.3	100.0	100.0	100.0	100.0	100.0	50.0	88.9	67.7
3	F	66.2	52.5	60.0	55.0	43.3	51.7	36.2	75.0	75.0	70.0	75.0	50.0	52.5	46.7	53.6
2	CS	85.0	80.0	20.0	70.0	80.0	60.0	45.0	100.0	100.0	40.0	80.0	20.0	70.0	53.3	66.1
0	CIS	78.3	46.7	53.3	53.3	31.1	48.9	28.3	73.3	73.3	66.7	66.7	53.3	54.4	35.6	51.1
0	М	72.7	50.0	27.3	63.7	51.5	48.5	47.7	81.8	81.8	63.7	72.7	27.3	50.0	40.0	54.0
4	F	88.9	61.1	66.7	50.0	33.3	55.6	13.9	77.8	77.8	55.6	66.7	66.7	68.5	40.7	55.9

2 <b>C</b>	s	80.4	28.6	28.6	100.0	67.1	71.4	54.3	85.7	100.0	50.0	71.4	85.7	73.8	60.7	69.6
0	IS	61.2	21.1	57.9	36.8	47.4	61.8	31.6	68.4	73.7	42.1	49.1	47.9	69.3	28.9	55.3
0 <b>N</b>	1	69.3	27.3	72.7	63.6	51.8	61.7	40.0	75.8	81.8	45.5	45.5	63.6	71.2	47.7	59.5
<sup>5</sup> F		64.2	20.0	63.3	46.7	53.3	66.7	48.0	71.1	80.0	43.3	62.2	66.7	70.0	30.0	58.9
2 <b>C</b>	S	37.5	75.0	29.2	87.5	57.5	68.7	57.5	100.0		50.0	81.2		42.5	53.1	63.7
0 <b>C</b>	IS	30.6	37.5	27.8	95.5	43.3	83.3	38.3	75.0		54.2	66.7		33.3	37.5	48.4
0 N	1	31.4	55.9	27.5	88.2	49.4	76.5	43.5	82.4		55.9	70.6		35.3	39.7	53.5
۶F		44.4	33.3	33.3	100.0	46.7	83.3	60.0	100.0		33.3	83.3		46.7	66.7	60.2
	<b>CS</b> = Computer Science <b>CIS</b> = Computer Information System															
N		= Ma	ale	F = 1	emale											
S	yn	= Sy	ntax	SDT =	Simple	Data	Туре									
S	S	= Sel	ection	Struct												
<b>FNT</b> = Functions <b>ARY</b> = Arrays <b>Str</b> = Struct /Records																
CLS = Classes STC = Stacks Que = Queue TR = Trees																
LNK = Linked List RCS = Recursion																
Prs = Problem Solving																
CMA = Computer Algebra OVL = Overall																

It should be mentioned that for 2003 and 2004 the number of programming questions were 10 per subject. Thus, the total number of questions was 30. In 2005 and 2006, the number of questions per subject was 20 and the overall programming questions were 60 per year. The total number of CIS female students who took the exit exam from 2003 to 2006 was 35 and male was 20. For CS major, the total number of female and male students was 11 and 17; respectively.

In the data presented in above table, some questions belonged to more than one category. For example, a question shown below requires the knowledge of both Linked List and Records.

Given the declarations

struct ListNode {
float volume;
ListNode\* link;
};
ListNode\* ptr;
what is the data type of the expression ptr ?
A. ListNode\* B. ListNode C. float D. \*ListNode
E. none--the expression is invalid

Thus, a correct answer to such a question indicates that the student probably possesses the knowledge of both topics.

Figures 1-17 shown in the appendix depict the graphical representations of the above data for each programming category during 2003-2006 based on student major and gender. Figure 18 shows the overall performance of students in the programming subjects of the exit exam. Figures 19 and 20 show the percentage of student population based on major and gender, respectively.

As indicated in Figure 1, students in both CS and CIS majors performed well (above 75%) in the Basic Computer Knowledge category. This is due to the fact that other courses covered similar contents. Figures 3, 6, and 8 show that students majoring in CS performed above satisfactory level while CIS majors did not. As it can been seen in Figure 12, both CS and CIS majors did not have a satisfactory performance in object-oriented programming topics.

This could be due to the fact that object-oriented programming involves more complex data types and structures. The figures also indicate that CS students generally performed better than CIS students in all categories except Basic Computer Knowledge and Tree Structure (Figures 1 and 17). In addition, Figure 18 indicates that the overall performance of CS majors was consistently better than CIS majors in the exit exams. This could be because of CS students are required to take more mathematics courses and thus they have a better logical thinking abilities and problem solving skills. Figure 20 shows no definite trends in the overall performance of male versus female students in the exit exams from statistical point of view. However, since the majority of female students were from CIS major with less mathematical background, the overall performance of female students could be considered better than male students.

## CONCLUSIONS

The following conclusions are drawn from the analysis of programming subjects in the exit exams given during 2003-2006 academic years.

- 1. Programming assignments should be given across the curriculum; especially in all Junior and Senior level courses. This may help students retain the programming knowledge and skills obtained in the first three programming classes on long-term basis.
- 2. Since students majoring in CIS did not generally perform as well as CS students, the mathematics components of CIS curriculum may need to be revised to include at least one more mathematics course such as Discrete Mathematics or Calculus I. It is well documented that mathematics courses could develop and enhance analytical, logical and critical thinking as well as problem solving skills in students [6, 8].
- 3. The programming subjects of the exit exam should be restructured to include a more balanced number of questions and coverage of programming materials. This will help improve statistics needed for each category to perform a more accurate analysis of data. In our exit exams given in 2003-2006 some categories included more questions than others.

#### REFERENCES

- McCracken M. "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students," Annual Joint Conference Integrating Technology into Computer Science Education, SESSION: ITiCSE, Canterbury, UK, 2001
- [2] Benford, S. D., Burke, E. K., Foxley, E., and Higgins, C. A. Ceilidh: "A courseware system for the assessment and administration of computer programming courses in higher education," Proceedings of the Interdisciplinary Workshop on Complex Learning in Computer Environments Joensuu, Finland, 1994
- [3] Wilson, B. C., & Shrock, "Contributing to Success in an Introductory Computer Science Course: A Study of Twelve Factors," The Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education, pp. 184-188, 2001
- [4] Dijkstra, E. W., "On the cruelty of really teaching computing science," Communications of the ACM 32(12) pp. 1398-1413, 1998.
- [5] Maheshwari, P. "Improving the learning environment in first-year programming: Integrating lectures, tutorials and laboratories," Journal of Computers in Mathematics and Science Teaching 16 (1) pp. 111-131, 1997.
- [6] Suken Dey, "Effects of mathematics preparation and prior language exposure on perceived performance in introductory computer science courses," ACM SIGCSE Bulletin, Vol. 18, Issue 1, pp. 144 -1448, 1986.
- [7] Terry R. Hostetler, "Predicting student success in an introductory programming course," ACM SIGCSE Bulletin, Vol. 15, Issue 3, pp. 40 43, 1983.

- [8] Lai, Shu-Ling; Repman, Judith L., "The effects of analogies and mathematics ability on students' programming learning using computer-based learning," International Journal of Instructional Media, Vol. 23, No. 4, pp. 355-364, 1996
- [9] D.S. Malik, "C++ Programming: From Problem Analysis to Program Design", Thomson Course Technology, 2002
- [10] N. Dale, D. Teague, "C++ plus Data Structures", Jones & Bartlett Publishers, 2003
- [11] L. Nyhoff "C++ An Introduction to Data Structures", Prentice Hall

#### Masoud Naghedolfeizi

Dr. Naghedolfeizi is an associate professor of computer science and engineering at Fort Valley State University. His teaching interests include computer-based measurement and instrumentation, data communications, computer networks, and programming languages. His current research interests include applied artificial neural networks, signal processing, and measurement systems.

#### Singli Garcia Otera

Dr. Garica-Otero is a professor of electrical and computer engineering in the Department of Mathematics and Computer Science at Fort Valley State University. She completed her Ph.D. in Electrical Engineering from the University of Missouri. Her teaching interests include computer organization, operating systems, programming languages, and circuits. She is interested in digital signal processing, biosensors, computer-based measurement and instrumentation and parallel processing.

#### Nabil A. Yousif

Mr. Yousif is an assistant professor of computer science at Fort Valley State University. His area of teaching interest includes database systems, system, analysis and design, computer programming, and business information systems. His area of research interest includes database and information system designs.

# Appendix



Figure 1: Percentage of Correctness in Basic Computer Knowledge



Figure 3: Percentage of Correctness in Syntax



Figure 5 Percentage of Correctness in Repetition Structure



Figure 2: Percentage of Correctness in Simple Data Type



Figure 4: Percentage of Correctness in Selection Structure









Figure 9: Percentage of Correctness in Array



Figure 11: Percentage of Correctness in Recursion



Figure 8: Percentage of Correctness in Computer Algebra



Figure 10: Percentage of Correctness in Records



Figure 12: Percentage of Correctness in Class





Figure 15: Percentage of Correctness in Pointer







Figure 14: Percentage of Correctness in Queue



Figure 16: Percentage of Correctness in Linked List



Figure 18: Percentage of Correctness in Overall







Figure 20: Percentage of Student Population in Gender