Bergen Community College

Division of Mathematics, Science, & Technology

Department of Computer Science, Engineering & Information Technologies

COMPUTER SCIENCE PROGRAM REVIEW 2015

A PROCESS FOR
SELF-EVALUATION AND
CONTINUOUS IMPROVEMENT

<u>Table of Contents</u>

Bergen Community College

PROGRAM: ________Computer Science__________________________________

PROGRAM REVIEW TEAM: ____David Wang__________________________

Others involved:
Prof. Emily Vandalovsky, Chair, Department of Computer Science, Engineering & Information Technologies (CEIT)

Dr. P.J. Ricatto, Dean, Division of Mathematics, Science, & Technology

Dr. Carmen Martinez-Lopez, Dean, Division of Business, Social Sciences and Public Service

Dr. William Mullaney, Vice-President of Academic Affairs

Ms. Tonia McKoy and staff, Office of Institutional Research

Prof. Chris Rigby, Computer Science Faculty, Department of Applied Technologies, SUNY – Orange Community College

DATE OF THIS REPORT: ________August 31, 2015________________________

PERIOD OF YEARS BEING REVIEWED: ___2010 - 2014_________________

OVERVIEW

The Computer Science (CS) Program is comprised of AS.NSM.COMP.SCI - Associate Science in CS and CERT.COMP.SCI - Certificate in CS. The Program prepares students to transfer into the junior year of a baccalaureate program in CS or a related area such as computer engineering and computer information systems. The degree program includes the required and elective courses of several programming languages, mathematics, database, system analysis and design, data structures and algorithms, and computer organization. Computer science occupations include software developers, database administrators, hardware engineers, systems analysts, network architects, information security analysts, and many other both exciting and in high demand positions.

CS students need to develop a broad foundation in concepts that will help them remain adaptable in changeable work environments. The CS Program addresses current and long-term trends in the marketplace. A number of our courses are available over the Internet as part of the College's distance learning programs.

Program learning goals for CS students are:

- Apply knowledge and skills to create algorithmic solutions to problems effectively and efficiently;
- Become competent in using a programming language to solve problems;
- Have the ability to use current techniques, skills, and tools necessary for the practice of the discipline;
- Understand the components of the computer and the way that they work;
- Analyze a problem, and identify and define the computing requirements appropriate to its solution.

One of College’s mission and goals is to educate a diverse student population in a supportive and challenging academic environment. The CS Program is fully supporting and implementing plans continuously for that goal.

SUMMARY OF SIGNIFICANT DEVELOPMENTS SINCE LAST PROGRAM REVIEW

The last time the CS Program conducted a program-wide review was in 2004. The current review team can only locate a paper copy (not a digital file) of last program review with the assistance from Dr. Forsstrom, Chair of Mathematics Dept. There have been several significant developments since the last program review, with particular emphasis on challenges identified by the previous team.

1. **Continuous curriculum development.** All courses have undergone significant changes as a result of new platforms, new operating systems, new applications, and new publishers' support materials. Class software was very expensive for CS students in early 2000; however, that problem has been alleviated thanks to open source software and free technologies and tools offered by corporations. In the last review, the team tried to find a way of offering excellent and inexpensive learning software for college and students. This challenge has been solved. C++ was used in CIS287 Object-oriented Programming. The previous team recommended Java, a more object-based programming language. We have used Java for CIS287 since 2009. CS students are benefitting from learning two mainstream programming languages: C++ and Java. There were many recommended actions not done, and they are addressed in this report.

2. **Building partnerships/internships/externships.** An initial grant-funded opportunity along with the science department enabled us to provide 10-week summer residencies for two summers at Brookhaven National Laboratory (2006 and 2007). Students from Computer Science, Physics and Information Technology formed small teams to engage in original research, mentored by scientists at Brookhaven, along with our own faculty mentors. The first summer residency (2006) resulted in the students creating original stereo 3-D visualizations of protein molecules. The second summer residency (2007) resulted in very interesting comparisons of the efficacy and efficiency of writing programs in C++ using graphics primitives libraries, compared to the same thing done using Java graphics libraries. C++ resulted in faster performance (only 3-4% on average); Java graphics libraries permitted much more rapid development (about 50% of the time needed to develop the equivalent code in C++). Another grant-funded opportunity (2008-2009) enabled us to work with about half a dozen colleges in the northeast US on open-source Android development (HFOSS: Humanitarian Free and Open Source Software development). The aim of the grant was to foster practical application development skills among college students in service of humanitarian needs. The particular project our students worked on (again a 10-week summer program) enabled them to develop a smartphone application that would take the place of an EMS worker's manual clipboard for recording events in real time. The objective was to create a tool that would be more legible and useful to Emergency Room staff and that could be sent on ahead to the local ER via web upload (saving precious minutes) before the EMS vehicle even arrived at the local ER. Students completed a working prototype of the app during the 10-week period and presented results at a colloquium of all the participating colleges, at Trinity College in Connecticut at the end of the program. It is significant to note that the Bergen Community College student team was the sole 2-year school represented among the schools participating.

3. **More cooperation and coordination with the Dept. of Information Technology on grants and curriculum development.** Recent administrative reorganization resulted in a new Dept. of Computer Science, Engineering, & Information Technologies, which will allow for a closer and more expanded relation with the Information Technology Program.

FOCUS ON STUDENTS

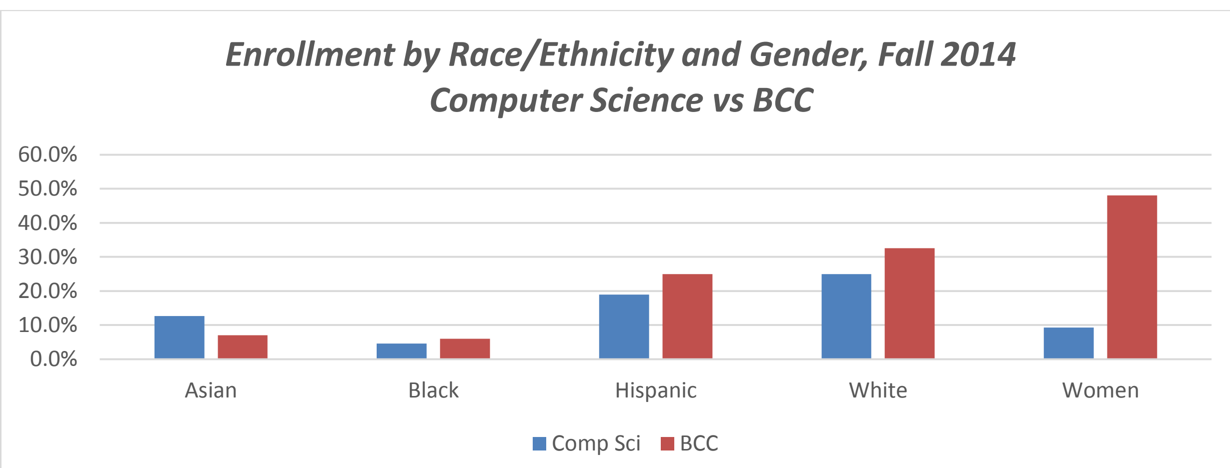
Demographics

Enrollment by Race/Ethnicity and Gender, Fall 2010 – Fall 2014

	Fall 2010		Fall 2011		Fall 2012		Fall 2013		Fall 2014	
	#	%	#	%	#	%	#	%	#	%
Am. Indian/Alaska Native	1	0.5%	1	0.4%	2	0.8%	0	0.0%	1	0.3%
Asian	32	17.1%	19	8.4%	29	11.9%	42	15.2%	41	12.7%
Black/African American	7	3.7%	10	4.4%	7	2.9%	12	4.3%	15	4.6%
Hawaiian/Pacific Islander	0	0.0%	1	0.4%	1	0.4%	0	0.0%	2	0.6%
Hispanic, all races	48	25.7%	61	27.0%	57	23.4%	55	19.9%	62	19.2%
Two or more races	1	0.5%	4	1.8%	6	2.5%	6	2.2%	8	2.5%
White	46	24.6%	56	24.8%	56	23.0%	66	23.9%	81	25.1%
Total Known Race	**135**	**72.2%**	**152**	**67.3%**	**158**	**64.8%**	**181**	**65.6%**	**210**	**65.0%**
Non-Resident Alien*	27	14.4%	28	12.4%	35	14.3%	39	14.1%	41	12.7%
Unknown	25	13.4%	46	20.4%	51	20.9%	56	20.3%	72	22.3%
Total Enrollment	**187**	**100%**	**226**	**100%**	**244**	**100%**	**276**	**100%**	**323**	**100%**

**Category includes individuals of diverse ethnic/racial backgrounds*

	Fall 2010		Fall 2011		Fall 2012		Fall 2013		Fall 2014	
	#	%	#	%	#	%	#	%	#	%
Male	173	92.5%	212	93.8%	228	93.4%	246	89.1%	293	90.7%
Female	14	7.5%	14	6.2%	16	6.6%	30	10.9%	30	9.3%
Total Enrollment	**187**	**100%**	**226**	**100%**	**244**	**100%**	**276**	**100%**	**323**	**100%**



With much more research needed to be done, NSF projected that nationwide only 13% of CS undergraduate majors are minorities (excluding Asian) and 10% are women. Except for Asian, the % of all other races in CS enrollment lags that of college wide student enrollment. The % of minorities in CS program at BCC is much higher than the projected nationwide rate and the rate of women is comparable to the nation-wide rate. We certainly can serve better to our students by working with STEM grants to promote CS among minorities and women.

Placement for First Time Students, 5 Years

	2009FA	2010FA	2011FA	2012FA	2013FA	All Terms
College Ready	12	7	16	14	23	72
	32.4%	14.9%	32.7%	26.9%	31.1%	27.8%
Remediation	23	37	30	37	43	170
	62.2%	78.7%	61.2%	71.2%	58.1%	65.6%
Unknown	2	3	3	1	8	17
	5.4%	6.4%	6.1%	1.9%	10.8%	6.6%
Total	**37**	**47**	**49**	**52**	**74**	**259**

Only less than 30% of first time CS students are college-ready. This imposes a huge challenge for students in term of CS program math requirement and prolongs their study time.

Student Satisfaction

In December 2013, a survey was sent to FY2012 and FY2013 graduates of Bergen Community College who transferred to a four-year institution according to data from the National Student Clearinghouse. The goal of this questionnaire was to garner feedback on the transfer experience of BCC graduates. In total, the survey was sent to 2,345 graduates who received an Associate's degree from BCC, whether it was an Associate's of Applied Science degree (AAS), an Associate's of Arts degree (AA), an Associate's of Fine Arts degree (AFA), or an Associate's of Science degree (AS). A web link to an online survey was sent out to the students' school and personal email addresses and 203 responses were collected. One Computer Science (AS) graduate's responses were collected – one was transferred to Montclair University with 60 BCC credits fully transferred. Due to the small sample size, the graduate's responses were reported on individually and numbered accordingly. Effective surveys which can attract a large number of responses from both current students and graduates are needed to be developed to assist the evaluation of student success. Surveys also can target the employers of BCC graduates.

Learning Outcomes Assessments

As of writing of this report, the only CS assessment report on file is for 2014-2016 academic year, which is attached below. We are continuing to search for the assessment reports from the previous cycles.

Bergen Community College

ASSESSMENT REPORT FORM FOR ACADEMIC PROGRAM

Assessment Period: Fall 2014 - Spring 2016
Department/Program: Computer Science/ Computer Science
Department Chair: Vacancy since 2013 and Starting in Spring 2015, Dr. R. Forsstrom
Department Assessment Liaison: David Wang
Date Submitted: December 15, 2014

- **Program Description or mission/goal statement of the Department/Program**:

The Computer Science Department offers an A.S. degree along with a certificate program. These programs provide fundamental knowledge in hardware, programming, database, system analysis and design, data structures and algorithms, and other computer science concepts.

The Computer Science Program strives to prepare the student for a transfer A.S. degree and/or a career in computer science. All programs provide a basic background in applications and concepts. Additionally, the Program provides a solid foundation for future study towards a four-year degree in computer science, computer engineering, computer information systems, and/or other relevant fields.

- **Program Learning Goals/Outcomes**:

Upon successful completion of the A.S. Degree in Computer Science, the student will be able to:

Foundations

1. Understand the fundamental concepts of computer science, be able to work with them, and understand how they can be used in real-world applications.
2. Be able to name the major components of a computer system and explain what each does.
3. Know the criteria to use in evaluating a software package and be able to perform tasks utilizing current problem solving software packages.
4. Be able to explain fundamental networking concepts.

Programming

1. Be able to design a program using structured development techniques.
2. Know how to implement a structured design using functions.
3. Understand the structure and processing of single and multi-dimensional arrays.
4. Know the fundamental algorithms for creating and processing sequential access files and random access files.
5. Be able to organize and represent data using structures.
6. Understand the fundamental techniques for sorting and searching data sets.

7. Be able to design a program using object-oriented techniques.
8. Be able to implement classes and their data and methods;
9. Be able to write programs using encapsulation, inheritance, and polymorphism concepts;
10. Be able to implement graphical user interfaces using swings, multithreading, and exception handling;
11. Be able to implement applets in HTML files.
12. Be able to use data structures.

Database

1. Be able to describe the components of a database system and the relationships between them.
2. Understand the major models of data and the role that each plays in the development process.
3. Know how to do the conceptual design of a database using the Entity-Relationship model of data.
4. Be able to use normalization theory to determine the quality of the design of a relational database.
5. Know how to use structured query language to create, modify, and query a relational database.
6. Understand the various schemes for the physical organization of a database.
7. Know the functions and goals of database administration.

Data structures and algorithms

1. Understand the concept of an abstract data type.
2. Be able to select the appropriate data structure and design the corresponding operations to implement an abstract data type.
3. Know the fundamental order of magnitude growth rates and how they are used to measure the run-time efficiency of an algorithm.
4. Know the principles of pointers, dynamic memory management, and be able to apply these concepts in constructing dynamic data structures.
5. Know the fundamental properties of stacks and queues and be able to implement them using dynamic linked-lists.
6. Be able to incorporate recursive techniques in the representation and implementation of an abstract data type.
7. Know the fundamental properties of binary trees, binary search trees, and general trees and be able to implement them using dynamic data structures.
8. Be able to represent graphs and networks using adjacency matrices and adjacency lists, and implement them using the appropriate data structure.
9. Know the major sorting algorithms and be able to identify the advantages and disadvantages of each.
10. Understand the use of hashing techniques in the storage and retrieval of data.

Hardware/Computer organization

1. Know how to trace the content of the appropriate special purpose registers in the CPU as assembly language instructions go through the instruction execution cycle.
2. Understand basic logic circuits, and understand how full adders are used to perform binary addition.
3. Know the different memory management techniques used by the operating system and the advantages and disadvantages of each.
4. Understanding how character data and real numbers are internally represented, and the ability to represent signed integers two's complement and in packed format.
5. Be able to identify the different addressing formats and addressing modes discussed in class, and will be able to write short code segments that use these addressing modes.
6. Be able to identify the parts of an Assembly Language program.

System analysis and design

1. Know the stages and sub-stages in the systems development life cycle.
2. Be able to apply PERT/CPM techniques to project management, monitoring and control: Gantt Charts.
3. Understand the tools and techniques for conducting a preliminary investigation of a systems project.
4. Know the advantages and disadvantages of the various techniques for information requirements analysis.
5. Be able to construct a conceptual model of a system using data flow diagrams, data specification tools, and process specification tools.
6. Be able to construct a conceptual of a system database using the Entity-Relationship model of data.
7. Know how to convert a conceptual database model into a relational database model.
8. Understand the fundamental concepts of systems design.
9. Be able to apply transform analysis and use design criteria to convert a data flow diagram to a system structure chart.
10. Know the various module specification techniques.

SEMESTER 1: CREATING PROGRAM-LEVEL ASSESSMENT PLAN

1. **Program Learning Goal(s) or Outcome(s) to be assessed (from the above section):**

Know how to implement a structured design using functions in programming.

2. **Means of Assessment:**

Two Projects.

Feedback from Dean: None and there is a Department Chair vacancy.

SEMESTER 2: DEVELOPING ASSESSMENT TOOL (s) and TIMELINE

3A. Describe or attach assessment tool (s), including sources of data, timeline for data collection and how data will be analyzed.

Will collect project#2 and project#3 from two sections of CIS265 - Advanced Programming Concepts offered in Spring 2015 by February, review/grade them by March, and analyze the data by April. Use a rubrics of comments (10%), prototype (10%), header(10%), and statements (70%) to grade each submitted project. Data will be aggregated to be analyzed and benchmarked with 3B.

3B. Desired results faculty would like to see.

70% of students who submit both projects are expected to have at least an average grade of 80% (see rubrics in 3A) from these two projects.

- **Feedback from CIE:**

SEMESTER 3: COLLECTING AND ANALYZING DATA

4. **Summary of Results (attach aggregated data table, survey tool, etc., to support the summary)**
5. **Recommendations for Improvement:**

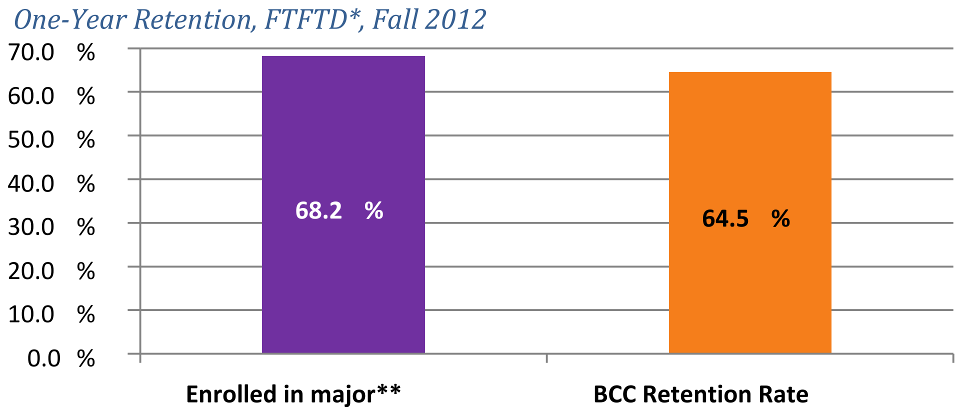
- **Feedback from Dean:**

SEMESTER 4: CLOSING THE LOOP AND SHARING KNOWLEDGE

6. **Use of Results:**

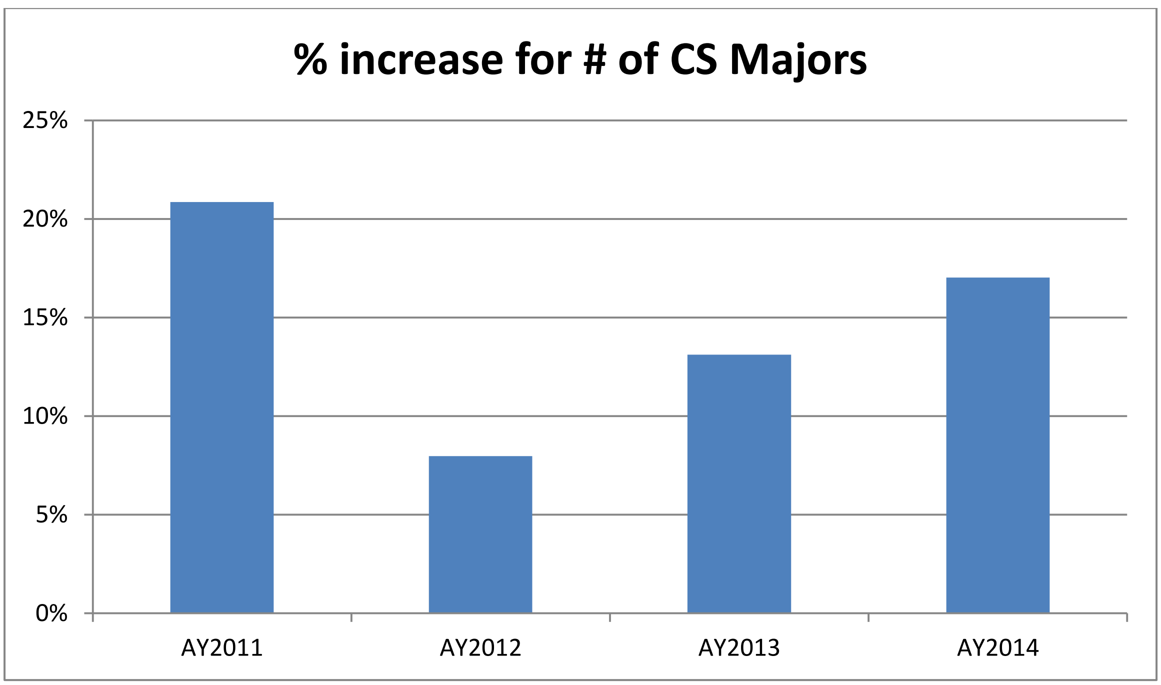
- **Feedback from CIE:**

Student Success

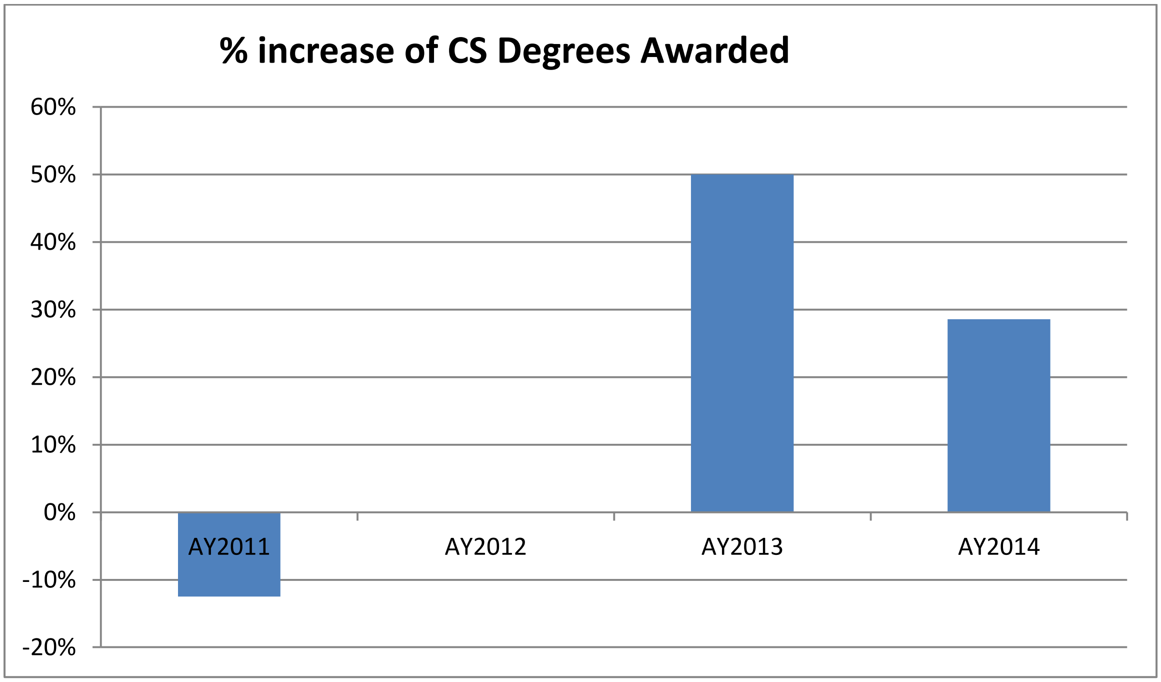


**One-Year Retention Rate includes First-Time, Full-Time, Degree Seeking Students in Fall 2012 who re-enrolled at BCC in Fall 2013. **Students who were enrolled in major in Fall 2012 and re-enrolled at BCC in Fall 2013. Source: SURE Enrollment File*

Career Program Name	Reporting Code	# of Declared Majors				
		2010	2011	2012	2013	2014
Computer Science	AS.NSM.COMP.SCI	187	226	244	276	323



Career Program Name	Reporting Code	Degrees Awarded					
		AY2010	AY2011	AY2012	AY2013	AY2014	5yr. Total
Computer Science	AS.NSM.COMP.SCI	16	14	14	21	27	92



While CS retention rate is comparable to college-wide rate, its enrollment and the number of degrees awarded for the past 5 years are in solid upward trend. This imposes both challenges and opportunities for us. How do we serve the highly increased enrollment with very limited faculty resources? Can we expand the enrollment ever more under the national priority for STEM education?

Data Needs

Effective surveys which can attract a large number of responses from both current students and graduates are needed to be developed to assist the evaluation of student success. Surveys also can target the employers of BCC graduates.

FOCUS ON FACULTY AND STAFF

Demographics/ Professional Activities

CS Program has one tenured faculty, one Lecturer, and 5-6 adjuncts. All of them have MS in Computer Science, Information Systems, and/or related fields.

David Wang, Assistant Professor			
Degrees	Certifications	Teaching Experience	Professional Experience/Affiliations
• Enrolled in Ed. Doctoral Program • MS Information Security & Assurance • MS Management Information Systems • BS Accounting/Computer Information Systems		• 8 years Community College	• 10 years software development

Recent Publications:

"Utilize the Case Study Approach in 200-Level Database Class: Challenges and Rewards", Tri-State Best Practices Conference "The Community College as a Place of Transition", BCC, Meadowlands Campus, NJ, March 9, 2013.

"Outside Classroom Learning: Developing a Mobile Phone Application – a Truly Fun Experience for Both Students and Faculty", 10th Faculty of the Future Conference, Bucks County Community College, PA, June 1, 2012.

"Intermediate-Level Open Source Software Development", workshop was given to College Community at Paramus Campus on March 27, 2012.

"Group Project Assignments in CIS265 – Advanced Programming Concepts", Tri-State Best Practices Conference "Innovation and Creativity in the Community College Classroom", BCC, Meadowlands Campus, NJ, February 18, 2012.

Grant/research activities:

NJ DOE Perkins Grant, 2011-2015. Wrote computer science departmental proposal and developed a series of workshops for College Community on open source software and mobile application development.

Humanitarian Free and Open Source Software (HFOSS) Grant co-sponsored by NSF and BCC, 2010- 2011. Worked with Prof. Madden, PI, and Prof. Vandalovsky to lead students on HFOSS development and published a presentation.

Affiliations:
Association Computer Machinery (ACM)/Computer Science Teachers Association (CSTA)
Professional Association for Database Developers
Community College Computer Consortium of NJ

George Chudyk, Lecturer			
Degrees	Certifications	Teaching Experience	Professional Experience/Affiliations
• MS Computer Science • BS Industrial Management	ITIL – project management Many IBM technical courses throughout my career	• 14 years College • 5 years K12	• 26 years professional IT full time – UPS. Including Methods Analyst, Application Programmer, User Representative, Systems Programmer, Systems Manager.
Affiliations: IBM Academic Initiative - member			

Michael Zobbi, Adjunct Instructor			
Degrees	Certifications	Teaching Experience	Professional Experience/Affiliations
• MS Computer Science • BS Computer Science	CompTIA A+	• 4 years College	• 4 years as Applications Developer at UPS (Financial Systems support Oracle R12 E-Business Suite) • 1 year at Merck & Co. as Business Analyst (Working with business and IT groups)

Alex Luy, Adjunct Instructor			
Degrees	Certifications	Teaching Experience	Professional Experience/Affiliations

• MS Computer Science • BS Computer Science	PMP (Project Management Professional) CSSLP (Certified Secure Software Lifecycle Professional)	• 7 years Community College	• 12 years IT Professional (developed, designed websites, created web services and trained clients; Architected databases;Agile Technical Project Manager;Technical Expert in all phases of the software development lifecycle.
Publications: https://www.linkedin.com/pulse/tech-interview-college-level-alex-luy-pmp-csslp **Affiliations**: PMI, ISC2			

Alan Eliscu, Adjunct Instructor			
Degrees	Certifications	Teaching Experience	Professional Experience/Affiliations

• MS Computer Science
• BS Computer Science
		• 2 years Community College
		• Taught individual IT courses for continuing education departments of:
		  • SUNY Purchase
		  • Fordham University
		  • NYU
			• 34 years independent IT consultant (business analyst – full project life cycle; systems analyst; data architect; database administration)
			• 4 years employed with small consulting firm (similar experience as above)
			• 7 years employed with major food manufacturing and distribution company (application development; DB systems programmer; database designer)

Faisal Aljamal, Adjunct Instructor			
Degrees	Certifications	Teaching Experience	Professional Experience/Affiliations
• MS Computer Science • BS Computer Science	Microsoft Certified Systems Engineer	• 19 years Community College	28 years of IT experience • Software Asset Management • Release Management. • Desktop support. • Mainframe operationsprogrammer; database designer

Adjunct Faculty

Currently Prof. David Wang, CS liaison, regularly communicates with/mentors/recruites adjuncts and conducts their class visitations and evaluations.

Staff

One secretary is shared with Dean's office and Information Technology Program.

FOCUS ON CURRICULUM

Summary of Program Curriculum

The Associate in Science Degree Program in Computer Science prepares students to transfer into the junior year of a baccalaureate program in computer science or a related area such as computer engineering or computer information systems. The certificate program is primarily intended for individuals who have completed a degree in another discipline and would now like to transition into the field of computer science or a related area such as computer engineering or computer information systems. The CS Program includes required and elective courses in current programming languages, discrete mathematics, data structures and algorithms, database systems, system analysis and design, and computer organization. Computer science occupations include teaching positions at all levels, software developers, database administrators, hardware engineers, systems analysts, network architects, information security analysts, and many other both exciting and in high demand positions.

Program learning goals for CS students are:

- Apply knowledge and skills to create algorithmic solutions to problems effectively and efficiently;

- Become competent in using a programming language to solve problems;

- Have the ability to use current techniques, skills, and tools necessary for the practice of the discipline;

- Understand the components of the computer and the way that they work;

- Analyze a problem, and identify and define the computing requirements appropriate to its solution.

CS certificate curriculum map link:
http://www.bergen.edu/Portals/0/Docs/CIE/Assessment/Assessment%20of%20Student%20Learning/Curriculum%20Maps/Fall%202014/FA14_Computer%20Science_CERT.COMP.SCI.pdf

CS degree curriculum map link:
http://www.bergen.edu/Portals/0/Docs/CIE/Assessment/Assessment%20of%20Student%20Learning/Curriculum%20Maps/Fall%202014/FA14_Computer%20Science_AS.NSM.COMP.SCI.pdf

CS Master course syllabi link: http://www.bergen.edu/academics/syllabi-central

Curricular Issues

In September 2008, New Jersey signed into law a Comprehensive State-wide Transfer Agreement. This Agreement, which is also known as the Lampitt Law, is designed to facilitate a smooth transfer from a New Jersey community college to the New Jersey public four-year colleges and universities. The Agreement provides for full transfer of the Associate in Arts (AA) and Associate in Science (AS) degree credits for community college graduates provided, the student plans to enroll in a coordinating major at the upper division college/university and, the student has followed certain recommended guidelines as are posted on NJ Transfer website. More specific articulations with 4-year New Jersey colleges/universities are to be developed

All courses have undergone significant changes as a result of new platforms, new operating systems, new applications, and new publishers' support materials Class software was very expensive for CS students in early 2000 but that is no longer the case now thanks to open source software and free technologies and tools offered by corporation. In the last review, the team tried to find a way of offering excellent and cheap learning software for college and students. This challenge has been solved. C++ was used in CIS287 Object-oriented Programming. Previous team recommended to us Java, a more object based programming language. We have used Java for CIS287 since 2009.

Lead-in Courses

Thanks to the STEM "GPS" Grant, for the first time, boot camp/prep classes were offered in the summer of 2015 to prepare students for CIS158 Introduction to CS /CIS165-Fundamentals of Programming. CS courses also enroll a number of local HS students.

Scheduling

CS enrollment pattern has been upward in recent years thanks to job outlook and future technology innovation. CS Program offers regular, DL, and hybrid classes to meet student needs. Most classes reach their full capacities. In recent years, some of the CS major required classes have been in high demand and we had to add one additional section to meet enrollment growth.

Assessment

The assessment report on file only started for the first time in 2014. The data analysis is highly limited at the point. We can improve this process greatly with the systematic assessment plan/cycle's rolling out.

Innovations or Changes in Last Five Years

There are several trends and development in CS: cyber security, social e-commerce, mobile applications, and many others. If we have the resources, we can certainly develop some open electives and/or workshops on those topics for students.

FOCUS ON SUPPORT

Technology, facilities and equipment. With respect to technical infrastructure and technical support, the College's support for the CS Program is quite extensive. The IT staff works diligently to make sure that the computer laboratories used by CS students are set up, imaged and configured appropriately to meet our needs. Due to the nature of CS Program, our students need dedicated computer labs as many 4-year colleges have been doing so. Following the merger with Information Technology, the College can set aside dedicated space for use by students majored in CS and Information Technology.

Learning Resources. Resources and support are also available through the College's Center for Innovation in Teaching and Learning (CITL), which provides general instructional support services and which has been particularly helpful with technical online course support and with creative ideas for how to best use technology (both as an end and as a means) in the classroom. The library maintains a good collection of CS-related paper books. Digital copies are much preferred if publishers make them available.

Marketing and Public Relations. The CS Program has worked in the past with the Public Relations department to produce printed brochures. The faculty attends every Admissions Open House in Paramus. All of these events have been important in the recruitment and maintenance of our student populations.

Support Services. All tenured and tenure-track faculty are required to perform 14 hours of advising per semester. Currently, we have only one faculty doing that service.

Resources and Budget. The substantial growth in the enrollment and graduation numbers for the Computer Science degree provides evidence that the discipline is in need of more tenure track faculty to cover all the sections currently offered. The additional instructors will also allow the offerings to grow to meet the enrollment demands of the future. The growing CS discipline which currently has a discipline liaison who receives 4.0 hours of administrative re-assign annually needs to be given more administrative support so that this individual in concert with the CEIT Department Chair can better serve the needs of the computer science students.

FOCUS ON COMMUNITY

Community Groups

Annually, the CS Program attends 'Compute@BCC' conference targeted at high school technology coordinators, technology faculty, computer science faculty, and high school administrators, as well as colleagues at our sister institutions (New Jersey community colleges) and 4-year institutions. These half-day events, presented in conjunction with the Computer Science Teachers Association (a national professional organization primarily composed of high school science teachers) and the Community College Computer Consortium, provide sessions where participants can gain insight into how to make what they do with computing technology at the high school level more compelling and engaging for students, can learn about some practical, easy-to-use (and mostly free) tools they can use right away in their classrooms, as well as learn about the latest developments in K-12 Computer Science educational standards being developed nationwide (more on this in the next paragraph). We have averaged about 70-80 attendees at these events, representing some 40-45 institutions across NJ, PA, NY and CT. Information Technology Dept. was the leading organizer.

With the funding from Perkins grant, CS Program also conducted community-wide workshops on CS topics and its faculty development was mainly in the form of conferences. The Program regularly recommends qualitied CS tutors to BCC's learning center.

Community Issues Related to Program

The CS enrollment pattern is upward in recent years thanks to job outlook and the wide application of technology innovation. The Bureau of Labor Statistics (bls.gov) maintains an Occupational Outlook web site where users can look up specific occupation titles to see what the hiring projections are over the next several years. Current projections are for the period 2012-2022 (prior projections were for the period 2010-2020). In most job categories related to CS hiring projections have shown growth rates in the 'Faster Than Average' to 'Much Faster Than Average' categories (their two highest categories). This has been consistent since we first started tracking trends on this web site in 2009. The BLS web site also shows median salaries for all job titles listed. In most IT-related jobs median salaries are quite high, ranging from $72,560 for Network and Computer Systems Administrators to $93,350 for Software Developers. One outlier is the Computer Support Specialists who can expect a median salary of $48,900. These are, of course, median salaries and do not represent what our graduates can expect as new entrants to the labor market; however, when compared to other professions across the US economy, most of these median salaries fare quite well. Also, these are national figures. In the NY Metropolitan Area we would expect many of these figures to be somewhat higher, given that cost-of-living is higher than average in this region. There are tremendous opportunities of STEM grant funding from government and private sectors for

External Requirements or Considerations

Current CS Program curriculum follows the standard recommended by ACM Committee for Computing Education in Community Colleges. The curriculum is up-to-date and meets the transfer degree requirement.

Advisory Boards

There's no advisory board for CS Program. We are exploring the possibility of establishing one in the near future.

SUMMARY

Program Achievements, Progress Made Since Last Review

1. **Continuous curriculum development.** All courses have undergone significant changes as a result of the nature of CS. All courses utilize the latest main stream software which is also accessible to students free of charge.

2. **Building partnerships/internships/externships.** Grants from NSF, NJ DOE, and other private findings have provided tremendous academic development and learning opportunities for students. CS Program is accepted to HPCC Systems (a leader in open source, big data processing, and analytics) academic community to benefit students and faculty from training, invitation only conferences, and new technology exploration.

3. **More cooperation and coordination with the Information Technology Program on grants and curriculum development.** Recent administrative reorganization resulted in a new Dept. of Computer Science, Engineering, & Information Technologies, which will allow for a closer and more expanded relation with the Information Technology Program.

4. **CS enrollment increased to 323 in 2014 from 187 in 2010 with a solid annualized 18% increase over the last 5 years.** This imposes both challenges and opportunities for us. How do we serve our students from limited faculty resource? Can we expand the enrollment ever more under national priority on STEM education?

Mission/Goals/Objectives

Considering the number of current faculty to support over 200 CS majors, the CS Program meets its mission, goals and objectives very well.

Strengths

- Well-designed curriculum
- Dedicated and qualified faculty including adjuncts
- Students' motivation and quality
- Updated facility equipment and low cost of software
- 4-year college degree transferability
- Bright Job outlook
- National-level funding priority

Challenges

- Very low number of tenured/tenure-track faculty
- No dedicated computer lab spaces
- Funding for faculty development and external relationship development (ex, high school, local community out research, grant writing)

Recommendations for Change

- Increase the number of tenured/tenure-track faculty from one to at least three.
- Work with Information Technology on STEM grant
- Develop more specific articulation agreements with NJ 4-year state colleges
- Establish the CS advisory committee
- Expand careers services for students
- Designate dedicated computer lab spaces to CS majors
- Sufficient funding for faculty development and external relationship development (i.e., high school, local community out research, grant writing)

ACTION PLAN

Goal 1: Increase our tenure-track and tenured faculty to at least three.

Objective 1: Conduct 1 tenure-track search in Spring 2016
Timeframe: 2016
Responsible Parties: CS Search Committee, Dean PJ Ricatto, VPAA William Mullaney
Resource Implications: the College must be willing to support one additional tenure-track faculty salary
Objective 2: Conduct 1 tenure-track search in Spring 2017
Timeframe: 2017
Responsible Parties: CS Search Committee, Dean PJ Ricatto, VPAA William Mullaney
Resource Implications: the College must be willing to support 1 additional tenure-track faculty salaries so in 2017 we'll have 3 tenured/ tenure-track faculty to support over 200 CS students. This is the key and prerequisite to realize the following goals.

Goal 2: Designate dedicated computer lab spaces to CS majors. Ideally make two computer labs on the second floor in Tech Building as the home labs reserved for CS classes and CS students.

Timeframe: 2017
Responsible Parties: CS Faculty, Chair Emily Vandalovsky, IT, Dean PJ Ricatto, VPAA William Mullaney
Resource Implications: Fairly minimal space allocation issues following the move of the Health Sciences programs to the new Health Sciences Building. In the near term (2-3 years), CS students will have their own specialized labs to explore latest technologies and develop computer applications.

Goal 3: Work with career services and regional employers to expand internships.

Objective 1: increase number of internships to 15 per semester by Fall 2017.
Timeframe: 2017
Responsible Parties: CS faculty, Chair Emily Vandalovsky, Career Service Dept., IT
Resource Implications: Some students will be working at IT Dept. at BCC instead of outside employers. Thus it will save the College an estimate of $50000.

Goal 4: Develop more articulation agreements.

Timeframe: 2017 – 2108
Responsible Parties: CS faculty, Chair Emily Vandalovsky, Dean PJ Ricatto, VPAA William Mullaney, Transfer Coordinator Dianna O'Connor
Resource Implications: More informed and careful advisement will be needed going forward. It is recommended that the College provide additional support for transfer counseling (which in our view is currently understaffed).

Appendix A
External Program Reviewer's Report by Christopher Rigby

Need / Demand for the Program

It is customary to begin program reviews of this type with a section outlining the need or demand for the program. Two year Computer Science programs like Bergen's provide a vital link in the educational chain for technology graduates. It is not controversial to note that technology jobs are generally on the rise, especially in larger metropolitan areas. This can be demonstrated by examining recent NJ Department of Labor Employment Projections (2012-2022) [see: http://lwd.dol.state.nj.us/labor/lpa/employ/indoccpj/nj_ind_occ_projections_12_22.pdf]. Although statewide "Information" jobs may face a very slight decline (a loss of less than 1%), the "Northern Region" of New Jersey is projected to have very robust job growth (up to 8%), the largest projected growth rate in the state (nearly double the number of jobs created than other regions of the state). Bergen County alone is projected to have the most annual average job openings in the state. In addition, proximity to the wider pool of jobs available in the New York metropolitan area will only serve to increase employment prospects for motivated graduates.

Program Structure

The program review document presents a very strong program, particularly from the viewpoint of academic structure. The program sequence is generally informed by the guidelines laid down for two year Computer Science programs by the Association for Computing Machinery (ACM) [see: https://www.acm.org/education/CS2013-final-report.pdf], the primary academic organization for Computer Science. This seems wise given the recent development of New Jersey's Comprehensive Statewide Transfer Agreement policy. Program standardization is a significant step toward adapting to the new statewide regulations, and Bergen's adoption of ACM guidelines shows their dedication to this important goal.

The SUNY system recently went through a similar process, dubbed Seamless Transfer, requiring each Community College campus to align their Associate of Science programs to the corresponding programs at the four-year schools. All 64 units of SUNY sent delegates to form committees for each affected program (I actually served as a member of the Computer Science Seamless Transfer committee). The CS committee ultimately produced the following four course core sequence: Computer Science 1, Computer Science 2, Data Structures, and Computer Organization and Assembly Language. In addition, the committee recommended two terms of Calculus, Discrete Mathematics, and a two semester science sequence (preferably Calculus based Physics). This mirrors almost exactly the program structure of Bergen's Computer Science program, which is remarkable given the lack of centralized guidance in the New Jersey system.

A brief survey of similar Computer Science degree programs in comparable institutions in the adjacent area (County College of Morris, Essex County College, Passaic County Community College, etc.) shows how similar this program is to surrounding colleges in the New Jersey state system. This suggests both the influence of market forces on the selection of a standard set of skills appropriate for such a degree, as well as the influence of the ACM standards on two-year transfer programs of this nature. In short, the course sequence provides a firm set of foundational skills that will serve students well in their transfer institutions as well as in industry. The current curriculum ensures student success.

As to course content, even a cursory examination of the existing course syllabi (which Bergen appropriately makes common to all sections of each course), shows that each course covers the necessary topics in appropriate depth, and adhere to basic ACM guidelines. Similarly, the text books selected for each course are recent and relevant, and the chapters utilized for each class reflect the proper topics that should be covered in each course (We even use several of the same textbooks at my branch of SUNY). In addition, the program strikes a sufficient balance of classroom instruction with laboratory assignments and hands-on projects. Reviewed syllabi show the density of material covered, and display a refreshing predilection to reinforce classroom lectures with a variety of demanding laboratory projects and hands-on assignments.

Similarly, the choice of language for programming instruction is a sound one. Some two-year institutions select a language based solely on the requirements of transfer institutions (generally Java). Bergen's program begins instruction with C++, a choice that will serve students well in later courses such as Operating Systems and Systems Programming. As noted in the review document, previous reviewers suggested they switch the instructional language to Java. Bergen has complied by providing instruction in Java in their Object Oriented Programming course, thus giving students experience with two widely used high level programming languages.

Instructional staff

Teaching staff are more than adequately qualified, with appropriate terminal degrees reinforced by many years of industry experience. The instructors have a wide range of areas of expertise, ranging from programming and web technologies to databases and system design. However, the entire staff currently consists of only a single full-time, tenure-track instructor, with the remainder of the staff populated by (admittedly exceptional) adjunct instructors.

Relying on a sole full-time instructor for a program of this size is problematic for the program for a variety of reasons. Enrollment numbers indicate a larger full-time faculty component. Similarly, Computer Science is a tremendously broad field, which today touches on almost every other area of academic and scientific endeavor. As such, it consists of many distinct specialty areas and no one faculty member can master them all. It is preferable to have a number of different voices to create a balanced and fully rounded program. Adding to the breadth of faculty resources brings the diversity of views necessary not only for appropriate curriculum development and course design, but also to provide varied perspectives and specialties in the classroom. These concerns highlight the need for additional full-time, tenure-track instructional staff.

Adding further faculty resources will also resonate beyond the classroom. The program currently does not have sufficient staff to provide necessary advising hours for such a large student population, instead depending on generic advisement from the advising center, which is not optimal for student success. Similarly, since the sole faculty member is overburdened, there is little time for support of extracurricular activities like clubs. In fact, this was the major complaint expressed by the students – they regretted the lack of faculty leadership for a dedicated Computer Science club. Such clubs are a vital part of any technical department, encouraging student assimilation and offering students a necessary level of support and community (peer-tutoring, programming contests, etc.). Adding more full time faculty will provide sustenance for a supportive and well-integrated program.

Physical Infrastructure and Facilities

Computer Science and Technology students have a particular set of computational needs, which differ drastically from the average student. Whereas the average student merely needs access to an Internet browser and an Office suite, technology students require access to various different operating systems, compilers, and database suites which may need to be reconfigured between classes or over the course of the semester. As such, for any Computer Science or Technology program it is recommended that an institution dedicate specific laboratory resources to meet these particular needs.

In addition, such labs may necessitate allocating additional staffing resources for installation and maintenance. Knowledgeable professionals must be available to select and install software, prepare laboratory environments, image machines, etc. Much of this work can be obviated by the utilization of a virtualization environment – for example, at SUNY Orange we use OpenStack. However, the hardware necessary to run virtual machines may be more expensive than standard lab machines, a factor to be considered when specifying the laboratory environment (VirtualBox is a viable "resident" alternative). Either way, designated IT/IS staff (or significant release time if this work is to be performed by faculty members) are indicated to maintain a new laboratory environment.

Program Goals and Objectives

The foundational program learning outcomes discussed in the department assessment (page 8) seem appropriate, and grounded in topics suitable for such a program. As noted above, the adherence to the main ACM program objectives ensures that these goals will align well with both four-year institutions and industry. Individual objectives and goals are reflected in the corresponding course syllabi. For example, the learning objectives for computer programming (listed on pages 8 and 9 of the program review document) are more than adequately met in the topics covered in the core programming sequence (and the Object Oriented Programming elective). Many of the subsets of objectives are mapped directly to particular courses (for example, Hardware/Computer organization to CIS 271 – Computer Organization and Assembly Language; System analysis and design to CIS 289, etc.). In short, the core sequence seems designed to meet the designated objectives and goals of the program, and ensure that courses introduce and reinforce appropriate topics to students.

"Assessment" has recently become a watchword for educational institutions, particularly those in the process of Middle States accreditation. Fortunately, Bergen's program is in the middle of carrying out a program-level assessment plan, gathering evidence concerning the extent to which current courses reflect the stated program learning outcomes. Their current plan assesses the extent to which courses teach students to implement a structured design using functions

(Program Learning Outcomes number 1 and 2 for the Programming sequence). The program is currently in the second of the four semester assessment sequence, so we do not yet have data to assess their success. However, examination of syllabi and course materials allows us to infer a reasonable level of success.

Discussion with students about the academic rigor and execution of the courses also met with a great deal of encouragement. Students had very few negative comments about the program, and generally viewed it quite favorably, with the majority of responses being overwhelmingly positive. Many stressed the demanding nature of the assignments which, viewed from a wider perspective, seem entirely appropriate.

This might seem a good point to note that the program currently lacks a strong advisory board. Such a committee is a recommended component of any robust Computer Science program, bringing a wider academic and industry perspective on a variety of issues, particularly current technology trends. The report recognizes this to be the case, and lists the establishment of an Advisory Board as one of its “Recommendations for Change” (page 24).

In summary, the course and laboratory materials seem to provide ample support for the general goals and learning outcomes of the program.

Instructional modes

Bergen’s program currently offers early core curriculum classes in a variety of instructional modes – traditional daytime classroom instruction, evening classes, and hybrid courses. Several of the courses, however, seem only to be offered in a daytime classroom format, which may ultimately cause problems for students who require other modes (for example, those who work full time) to complete the entire sequence for graduation. Of course, offering such courses depends ultimately on such factors as faculty availability and loading, which offers another argument for additional full time faculty.

Computer Science is a laboratory based discipline, requiring many hours completing practical coding assignments to successfully grasp theoretical topics. It is worth noting that currently Bergen’s laboratory based classes do not have separate lab sections. However, this is not necessarily a disadvantage, allowing instructors to adapt limited classroom hours to either lecture or lab as the topic dictates, offering students an optimal distribution of lecture / lab material.

Graduation Rates

A cursory examination of the graduation rates for the program might suggest that they are a trifle low. However, recall that generally two-year programs have lower rates. For example, recent SUNY statistics show that on average two-year programs have substantially lower rates than corresponding four-year programs. Also, Computer Science programs tend to have lower rates than many other programs. A variety of factors may be responsible: one is the difficulty of the discipline itself (particularly the stringent mathematics requirement); another is the higher percentage of two-year program applicants requiring remediation (again, particularly in mathematics, which may make it difficult to graduate in two years); finally, there is generally a high attrition rate due to students moving on to four-year schools (or even industry) without completing the degree. The rates posted in the report seem in line with the similar programs in the SUNY system (at least, those with which I am familiar).

The retention rates seem fine, again given the draw of four-year schools and employment. Better statistics on transfer rates (and particularly on specific transfer institutions) would be welcome. Ultimately, retention, transfer, and graduation rates do not seem inappropriate for a program of this size and type.

Articulation Agreements

Being essentially a transfer degree, graduates with an A.S. in Computer Science are generally bound for a four year institution. As such, articulations are a major benefit for programs of this type as they guarantee a level of equivalence between programs and generally require less duplication of coursework when the student reaches the transfer institution. Thus it is highly appropriate that Bergen has made one of its major goals the development of articulation agreements with appropriate transfer institutions (Goal 4, page 25).

Centralized programs, like the Comprehensive State-wide Transfer Agreement, may be of assistance here, providing crucial guidelines concerning appropriate curricula. However, as noted above, the current structure of the Computer Science program will make it attractive to four year institutions who will appreciate the strong foundational skills graduates of Bergen's program have collected.

Coordination with other academic units

It is beneficial that, at Bergen, Computer Science is embedded in larger Technology department. From a logistical viewpoint, this allows the sharing of technical and faculty resources, fosters the optimal allocation of educational assets, and nurtures the cross-pollination of knowledge that generally marks the technical disciplines. The sharing of faculty and educational resources with the wider Technology program might be beneficial for both degrees. For example, the Computer Science degree currently does not offer much in terms of instruction in Linux, whereas the Networking Administration Degree does. Adding some Linux based laboratories or courses would make the CS students more well-rounded and attractive to (and more successful at) transfer institutions and in industry. Thus it might benefit the program to take greater advantage of the confluence of resources that being embedded in the wider Technology program provides. However, it should be noted that this does not obviate the necessity of further designated full-time, tenure-track faculty in Computer Science.

Summary

As noted above, the Program Review report and campus visit portrayed a very strong program, well-structured and similar to programs at comparable neighboring institutions. The reviewer strongly supports the "Recommendations for Change" and "Action Plan" goals outlined at the end of the report (pages 24 and 25). Let us examine each in turn.

Goal 1 – for reasons noted above, additional full-time, tenure-track faculty is indicated for a robust program of this size. This goal should have high priority (as noted in the report's timeframe) and will have the greatest ultimate impact on the program. Hopefully the college can support such additional faculty to help this strong program reach its full potential.

Goal 2 – Dedicated laboratory resources (and associated support staff resources) are also indicated to help this program realize its goals. Recall that the computational needs of Computer Science students are by nature different from the general student population, requiring access to a wider array of operating systems and utilities.

Goal 3 – Computer Science is ultimately a practical discipline, so Internships can play a critical role in preparing students for higher level coursework and, ultimately, entry into industry. One might argue that students at this level have not yet amassed a set of skills that would enable them to function in such a capacity. However, early exposure to complex, real-world technology systems is the best path to ensure student success. Not only does it introduce students to the intricacy of such systems, it also encourages their creative potential in a way that routine coursework does not.

Goal 4 – Articulation agreements ultimately promote the health of Computer Science programs, streamlining the transfer process and facilitating student success at their chosen four-year institution. As such, the development of such articulations with appropriate transfer institutions should be a priority. Developing such relationships will not only benefit transferring students, but will allow Bergen's faculty to develop a better understanding of the requirements and foci of the four-year programs that Bergen ultimately feeds.

In summary, supporting the goals outlined in the report's "Action Plan" will help ensure the continued success of this program. Additional faculty will bolster the program's already considerable strengths while adding further expertise and broader perspective.

Appendix B
External Program Reviewer's CV

Christopher David Rigby

361 All Angels Hill Road ♦ Wappingers Falls, NY 12590 ♦ (845) 632-2227 ♦ chris.rigby@gmail.com

Accomplished Information Technology professional with more than a dozen years experience spanning the entire System Development Lifecycle. Expertise with software development in a variety of languages (C, C++, Python, Java), particularly on a Unix/Linux platform. Integration with traditional databases (particularly Oracle and MySQL) as well as with NoSQL-style distributed key-value stores (mongoDB, couchDB). Managerial experience with large software teams on substantial projects. Experience in all standard tiers, from back-end batch processing, through middleware and web-based front-ends. Copious work in scalability and integration, with experience in distributed and grid processing and virtualization in large-scale data environments.

Education

1998-2002 Graduate Center, City University of New York
Graduate coursework in both Computer Science and Anthropology Departments.
1998-2001 College of Staten Island, City University of New York
M.S. Computer Science
1992-1995 Brandeis University
Advanced graduate studies in Anthropology/Archaeology. Completed coursework and residency requirements for the Ph.D. (16 classes, including transfer credits).
1990-1992 Hunter College
Graduate work pursuant of a degree in Anthropology/Archaeology. (30 graduate credits). 1986-1990 College of Staten Island, City University of New York
B.A. Literature, B.A. Philosophy

Professional Experience

July 2015 – present ***Simons Foundation*** New York, NY
Consulting work for a scientific research foundation, primarily for their Autism Research Initiative. Currently developing a workflow system to automate the delivery and processing of psychological survey instruments and related genetic information using a Django / Celery platform. Provide database design and integration of relational (Postgres) and NoSQL database systems. 2010 – present ***Orange County Community College, SUNY***
Assistant Professor
Full-time, tenure-track appointment as an Assistant Professor of Computer Science in the Applied Technologies Department.
Courses taught include: Computer Science I and II (programming in Java), C++ programming, Database Systems, Unix/Linux,
HTML, Systems Analysis/Design, Visual Basic, Management Information Systems, Operating Systems, Data Structures,
Assembler Language, Networking 1, Networking 2. Received 2012 *Innovative Instructional Technology Grant* (IITG) to build an OpenStack cluster for college use. Currently sole administrator for the Grizzly release 20 node OpenStack system, which is used to run diverse laboratory environments for 8 different classes.
June 2013 – August 2013 **noodle.org** New York, NY

NLP Consultant

Consulting work for an education software company started by the founder of the Princeton Review. Integrated Natural Language Processing routines into the Noodle educational search engine. Developed software that performed part-of-speech tagging and parsing for natural language query input. Identified and information that would optimize query results (location, category, etc.). Incorporated semantic web APIs for term synonyms and query expansion. Expanded queries were formatted for and passed to customized Solr/Lucene search system.

October 2010 – June 2012 **On Demand Books (www.ondemandbooks.com)** New York, NY *Senior Software Developer / Consultant*

Consulting work for growing technology startup in the publishing industry (ODBs Espresso Book Machine was listed as one of Time Magazine's Inventions of the Year). Developed back-end and front-end software for Web-based content-delivery systems in a variety of languages (Python, PHP, etc.). System integration with databases using ORM (MySQL, sqlalchemy), including clustering and replication. Design search utility for digital catalog content system utilizing several open-source packages (Sphinx, Whoosh). Service-oriented development for back-end architecture using a variety of current technologies – cherrypy, memcached, mongoDB.

September 2001 – August 2010 **billingIT / Information Strategies Group / Tangoe** New York, NY *Vice President for Systems and Development*

Oversee the development of telecommunications software and web front ends in a variety of languages (C, C++, Perl, Python, PHP, HTML, etc.) on a Unix/Linux platform. Application development and integration with several different database systems (Oracle, MS SQL Server, MySQL). Provide technical direction and leadership for the development and support staff. Design and implementation of embedded systems for a variety of telecommunications equipment. Responsible for several major projects including a PBX-based CDR system (hardware and software design), call pricing software based on several different vendor price index databases (Valucomm, etc.) and taxation systems (Vertex, etc.), and a proprietary call accounting system. Experience with High Performance Computing, implementation of Sun GridEngine architecture for distributed, high volume data processing.

Directly supervise a diverse team of software engineers and system administrators working on a variety of projects. Specify hardware and software solutions; allocate technology budget.

March 2001 – August 2001 **SavantNet** New York, NY *Lead Developer*

Copious programming (Perl, Python, C/C++) and scripting (Bourne shell, Perl) duties spanning both the administrative and production environments (web-based applications, CGI scripts, proprietary network and system monitoring applications, systems programming, SNMP, etc.), primarily on Linux/UNIX platforms. Backend integration of databases and LDAP for internet projects. Development of client-server architectures utilizing socket-based application communication (HTTPlib, etc.). Design and implementation of embedded systems for telecommunications clients. Provide technical direction and leadership for development staff.

May 2000-February 2001 **SavantNet** New York, NY

Senior Unix/Network Administrator

Responsible for the installation, maintenance, and administration of UNIX servers and Cisco networking equipment for a moderately sized Business-to-Business Network Service Provider. Administration of a variety of internet services -- email (sendmail), web servers (Apache), DNS (bind), etc. -- on Sun and Linux platforms. Setup and maintenance of CVS revision control repositories for proprietary source code collections. Installation and configuration of a variety of Cisco routers.

Familiarity with SNMP garnered through application programming in several languages (particularly C++ and Perl). Administration and application integration for several database systems (MySQL, Postgresql, Oracle, etc.). Designed and implemented network security policies (TCP wrappers, ipchains, PIX firewalls). Supervise team of network engineers. Specify hardware and software solutions and maintain relationships with multiple vendors. 1998 – 2001

College of Staten Island, City University of New York

Lecturer/Adjunct Professor

Teaching position in the Computer Science Department (one semester full time with Lecturer status, the remainder part time with Adjunct Professor status). Classes taught included Introduction to basic software packages and the microcomputer (computer basics, Microsoft Office, etc.), Introduction to Computer Science (programming in the C and C++ computer languages - taught several times), Discrete Mathematics (cross listed in the Computer Science and

Mathematics Departments), Mobile Robotics. Programming classes included both classroom lecture and laboratory sections. Designed syllabi and created supplementary course materials that emphasized training for real-world solutions. Also held a part-time teaching position in the Continuing Education Department as an Instructor for the College's Cisco Certified Networking Academy, geared toward training students for the CCNA exam. Repeatedly taught all four semesters of the Cisco program. Also taught several other courses for Continuing Education: Introduction to Computer Programming (C/C++), Microcomputer Maintenance, Web Programming (HTML).

August 1997 – May 2000 **College of Staten Island, City University of New York**

Unix Administrator

Responsible for the maintenance and administration of all UNIX systems at the campus. Experience included (but was not limited to) the installation and maintenance of software packages, shell scripting, and creating and maintaining user accounts. Systems included clusters of SGI Indigo Workstations, numerous PCs run on a Linux platform, and a NIS+ based Solaris network populated with a variety of Sun equipment (Sparc 5, 10, 20, and LX Workstations, Ultra I and IIs). Particularly familiar with RedHat Linux and Solaris. Familiarity with parallel and distributed processing garnered through the installation and use of parallel processing packages such as Linda, PVM, and MPI, as well as graduate coursework in that area. Responsible for information security on the Unix network.

Training

2014 *Mirantis*

Completed week long intensive OpenStack Administration course.

1999-2000 *Cisco Networking Academy*

Completed three week intensive course in Cisco Networking, designed for Instructors teaching the CCNA (Cisco Certified Networking Associate) exam material. Certified to teach all four semesters of the CCNA training sequence. 1999 *RedHat Inc.*

Intensive course in RedHat Linux Systems Administration.

RHCE (RedHat Certified Engineer)

1998 *Sun Microsystems*

Essentials of System Administration. Course included basics of system administration, hardware and software installation, upgrading, creating and administering accounts, backups, etc

Professional Experience

July 2015 – present ***Simons Foundation*** New York, NY

Consulting work for a scientific research foundation, primarily for their Autism Research Initiative. Currently developing a workflow system to automate the delivery and processing of psychological survey instruments and related genetic information using a Django / Celery platform. Provide database design and integration of relational (Postgres) and NoSQL database systems. 2010 – present ***Orange County Community College, SUNY***

Assistant Professor

Full-time, tenure-track appointment as an Assistant Professor of Computer Science in the Applied Technologies Department.

Courses taught include: Computer Science I and II (programming in Java), C++ programming, Database Systems, Unix/Linux,

HTML, Systems Analysis/Design, Visual Basic, Management Information Systems, Operating Systems, Data Structures,

Assembler Language, Networking 1, Networking 2. Received 2012 *Innovative Instructional Technology Grant* (IITG) to build an OpenStack cluster for college use. Currently sole administrator for the Grizzly release 20 node OpenStack system, which is used to run diverse laboratory environments for 8 different classes.

June 2013 – August 2013 **noodle.org** New York, NY

NLP Consultant

Consulting work for an education software company started by the founder of the Princeton Review. Integrated

Natural Language Processing routines into the Noodle educational search engine. Developed software that performed part-of-speech tagging and parsing for natural language query input. Identified and information that would optimize query results (location, category, etc.). Incorporated semantic web APIs for term synonyms and query expansion. Expanded queries were formatted for and passed to customized Solr/Lucene search system.

October 2010 – June 2012 **On Demand Books (www.ondemandbooks.com)** New York, NY *Senior Software Developer / Consultant*

Consulting work for growing technology startup in the publishing industry (ODBs Espresso Book Machine was listed as one of Time Magazine's Inventions of the Year). Developed back-end and front-end software for Web-based content-delivery systems in a variety of languages (Python, PHP, etc.). System integration with databases using ORM (MySQL, sqlalchemy), including clustering and replication. Design search utility for digital catalog content system utilizing several open-source packages (Sphinx, Whoosh). Service-oriented development for back-end architecture using a variety of current technologies – cherrypy, memcached, mongoDB.

September 2001 – August 2010 **billingIT / Information Strategies Group / Tangoe** New York, NY *Vice President for Systems and Development*

Oversee the development of telecommunications software and web front ends in a variety of languages (C, C++, Perl, Python, PHP, HTML, etc.) on a Unix/Linux platform. Application development and integration with several different database systems (Oracle, MS SQL Server, MySQL). Provide technical direction and leadership for the development and support staff. Design and implementation of embedded systems for a variety of telecommunications equipment. Responsible for several major projects including a PBX-based CDR system (hardware and software design), call pricing software based on several different vendor price index databases (Valucomm, etc.) and taxation systems (Vertex, etc.), and a proprietary call accounting system. Experience with High Performance Computing, implementation of Sun GridEngine architecture for distributed, high volume data processing.