Assessment Report

Department/Program: Mathematics & Computer Science Chair: Douglas Salane

Degree /Minor/Certificate/or other Program: Computer Science and Information Security Major

Time Period Covered for this Assessment Review: Fall 2014 and Spring 2015

Assessment occurred in the following courses:

Course	Semester and Year
CSCI 271 Introduction to Computing and Programming	Fall 2014
CSCI 373 Advanced Data Structures	Fall 2014
CSCI 400 Capstone Experience in Digital Forensics/Cybersecurity I	Fall 2014
MAT 204 Discrete Structures	Fall 2014
CSCI 272 Object Oriented Programming	Spring 2015
CSCI 274 Computer Architecture	Spring 2015
CSCI 377 Computer Algorithms	Spring 2015
CSCI 379 Computer Networking	Spring 2015

Direct Assessment of Learning Goals (Please attach to the report a copy of each rubric used.)

For reference, the following are the learning outcomes for the Computer Science and Information Security major. Graduates will be able to do the following:

L1: Use and critically evaluate the variety of theoretical approaches that are relevant to Computer Science and Information Security.

L2: Use and critically evaluate the variety of practical/hands-on/research approaches that are relevant to Computer Science and Information Security.

L3: Analyze the quality of the programs in Computer Science and Information Security to enable graduates to be successful in the highly competitive high technology industries and graduate schools.

L4: Communicate effectively through integrating theory, research and policy in written reports and presentations.

L5: Understand the ethical considerations and statutory requirements computer professionals encounter as care takers of sensitive data and designers and developers of systems that can impact the well-being of individuals and organizations.

After listing the learning goal(s), insert *the **percentage** of students falling into each performance level* in the following chart. Your descriptors for the performance levels may vary, and if they do please substitute yours, but it's important to specify which level "meets expectations" for your program.

Course Learning Goal(s) Assessed	Course Number (semester)	Sample Size	Exceeds Expectations	Meets Expectations	Approaches Expectations	Does Not Meet Expectations
Understand and work with abstract mathematical structures (e.g., integers, relations, graphs and trees) and use them to represent discrete objects such as computer networks and cryptographic systems. (Supports program learning outcome L1)	MAT 204 F	35	7 (20%)	17 (48%)	8 (23%)	3 (9%)
Describe the process by which high level process is deconstructed into a sequence of atomic logical steps. Develop an effective algorithm correspondingly to solve the specific problem. (Supports program learning outcome L1)	CSCI 271 F	9	3 (33%)	3 (33%)	1 (11%)	2 (23%)
Design and implement; implement the algorithm using C++ programming language. (Supports program learning outcome L2)	CSCI 271 F	9	4 (44%)	5 (56%)	0 (0%)	0 (0%)
Write assignments consisting of design, documented code, and comments to enhance the readability of the program. Communicate effectively through integrating theory, research and implementation in real projects. (Supports program learning outcome L4)	CSCI 271 F	9	5 (56%)	4 (44%)	0 (0%)	0 (0%)

Describe the process by which a high level process is deconstructed into a sequence of atomic logical steps. Develop an effective algorithm correspondingly to solve the specific problem. (Supports program learning outcome L1)	CSCI 272 S	26	6 (23%)	11 (42%)	3 (12%)	6 (23%)
Design and implement; implement the algorithm using a higher level language such as C++. (Supports program learning outcome L2)	CSCI 272 S	25	16 (64%)	6 (24%)	3 (12%)	0 (0%)
Be able to understand that numbers can be represented in multiple forms. Be able to convert signed and unsigned integers from one system to another. (Supports program learning outcome L1)	CSCI 274 S	23	8 (35%)	11 (48%)	4 (17%)	0 (0%)
Understand and be able to use the functions of various hardware components of a computer, such as processor registers and memory by being able to write simple code in lower level language for a contemporary processor, such as MIPS. (Supports program learning outcome L2)	CSCI 274 S	20	6 (30%)	8 (40%)	3 (15%)	3 (15%)
Organize information in computer programs to provide an efficient high-performance computing. (Supports learning outcome L2)	CSCI 373 F	16	0 (0%)	3 (19%)	5 (31%)	8 (50%)
Students will demonstrate familiarity with the Unix (as well as Windows/MacOS X) environment and be able to use cybersecurity mechanisms to improve the defensive stance of the environment. (Supports program learning outcome L3)	CSCI 400 F	25	5 (20%)	8 (32%)	8 (32%)	4 (16%)
Students are able to process to analyze an algorithm theoretically and describe the process to compare and classify different algorithms according to their asymptotic growth rate of the time function. (Supports program learning outcome L1)	CSCI 377 S	23	5 (22%)	11 (48%)	6 (26%)	1 (4%)
Students will be able to optimize implementation by using appropriate data structures by using appropriate storing and processing of input data. Also they will be able to design and implement assignment to identify procedures that would slow down the execution time of a program. (Supports program learning outcome L2)	CSCI 377 S	23	7 (30%)	9 (40%)	6 (26%)	1 (4%)

Identify basic key principles, methods and practices required to build, manage and secure local area and intranet networks. (Supports program learning outcome L1)	CSCI 379 S	22	6 (27%)	6 (27%)	3 (14%)	7 (32%)
Recognize and explain the role key components of the TCP/IP protocol suite play in modern packet based computer networks and network applications. (Supports program learning outcome L2)	CSCI 379 S	22	4 (18%)	7 (31%)	3 (14%)	8 (36%)
Be able to use protocol analyzers to analyze network traffic and protocols. (Supports program learning outcome L3)	CSCI 379 S	22	3 (14%)	10 (45%)	5 (23%)	4 (18%)

Indirect Assessment of Learning Goals

Indirect assessment typically relies on surveys (Student Experience Surveys, NSSE, etc.), post-graduate outcome data, graduation and retention rates, grades, and a variety of other data. They may be used to assess particular learning goals or for more global assessment of the program. Indirect assessment should be part of every yearly review as a supplement to the direct assessment of learning. See attachment for examples of direct and indirect instruments.

Learning Goal(s)	Course or Program Based?	Sample Size, if Known	Instrument	Data
L3: Analyze the quality of the programs in Computer Science and Information Security to enable graduates to be successful in the highly competitive high technology industries and graduate schools.	Program		Institutional Research, Fall 2014 Fact Book	Computer Science and Information Security (BS) Enrollment Trends Year / Number of Majors 2010 / 156 2011 / 185 2012 / 215 2013 / 232 2014* / 354 (9/20/14) 2015 / 424 (6/19/15) *New major

Assessment Process How did you go about assessing student learning in your program?

(Describe briefly the assessment methodology: sample selection, assessment instruments, scoring process, and assessment design)

Direct Assessment

As in other quantitative disciplines, assessment should determine whether students are able to apply the principles, methods and practices learned in a course to solve problems. In Computer Science this often means students have to analyze a problem, formulate a mathematical model, and then develop and implement an algorithm to solve it. This activity occurs repeatedly in just about all courses assessed here and just about all courses in the major. Even in Discrete Mathematics, a course that provides much of the mathematical underpinnings for computer science, students must come up with a mathematical formulation of a problem and then identify and effectively apply the appropriate solution method. In addition, students must demonstrate that they can present and explain problem solutions.

Instructors of the assessed courses in these reports developed several methods for assessing learning outcomes. Examinations that require students to solve problems and explain solutions are one instrument used for assessment. A second key technique is the programming project. Projects are assigned routinely in courses throughout the program. Generally, assessment of a learning outcome employs two basic rubrics to determine the degree to which students are achieving expectations. The first is the student's demonstrated ability to solve problems of varying difficulty. The second is the degree to which the student can apply the appropriate principles, methods, practices and tools to solve a particular problem. The first rubric is used extensively when an examination is the assessment instrument. The second is appropriate when a programming project is the instrument.

The first rubric is illustrated, for example, in the Advanced Data Structures course, where students were presented with collections of problems of varying difficulty. Those meeting expectations could solve most problems including more difficult ones that required a thorough understanding of computer algorithms and programming techniques. Those not meeting expectations were unable to write elementary C++ instructions (such as dynamic memory allocation) or illustrate the operation of the most popular sorting algorithm (quicksort).

The second rubric is illustrated in the programming exercises. Here, students must formulate the problem as a program, identify the appropriate algorithms and produce a working computer program to solve the problem. Students who exceed expectations are typically able to carry out all steps of the process to completion. Students who cannot even formulate the problem correctly do not meet expectations. Those who do formulate the problem and who are able to describe a procedure to solve the problem are considered to be approaching expectation. Those who can, for example, use step-wise iterative refinement, to completely describe the solution process can be considered to meet expectations.

In the coming year, we will consider whether the rubrics and assessment process widely used at the College are appropriate for assessment of the new Computer Science and Information Security program. Besides having an assessment process that enables us to monitor and improve the major, the process should enable the program to gain professional certifications. (See, for example, ACM-1 and ABET.) Such certifications require an assessment process; however, the process takes a somewhat different form from the process used at John Jay, which appears to be aimed largely at the social sciences and humanities. In the field of computing, assessment typically focusses on a concept and the level of mastery of the concept (ACM-2). Three levels of mastery are defined: familiarity, usage and assessment (not to be confused with the formal process of course/program evaluation). Familiarity means the student can identify the concept and understands it. Usage means the student knows how to use the concept in a specific way, e.g., to develop computer program. Assessment means the student is able to use the concept in solving a problem and can demonstrate why the use of the concept is the superior solution approach. Most course learning outcomes already include concepts that should be covered in the course. We plan to examine how we can assess student achievement based on the aforementioned professional guidelines.

References:

Association for Computing Machinery (ACM). Assessment Tools for Computing Curricula, Available at http://www.acm.org/education/assessment-tools-for-computing-curricula/

Association for Computing Machinery (ACM) and IEEE Computer Society Joint Task Force on Computing Curricula (Dec. 20, 2013). *Curricular 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Available at http://www.acm.org/education/CS2013-final-report.pdf

Accreditation Board for Engineering and Technology (ABET). Criteria for Accrediting Computer Programs. Available at http://www.abet.org/DisplayTemplates/DocsHandbook.aspx?id=3148

Indirect Assessment

We used Trends in Major Enrollment in the Fall 2014 Fact Book compiled by the Office of Institutional Research to determine if there was renewed student interest in the new Computer Science and Information Security (CSIS) program. The 2014 and 2015 counts of the majors were derived from the lists of majors as reported in CUNY First. This assessment is related to Program learning goal L3 since the major provides excellent preparation for employment in the IT field and graduate study in a various fields that rely on quantitative and computing skills.

Although the CSIS major has been a relatively small major, the data show interest is growing extremely rapidly. For the revised major that began in Fall 2014, we plan to develop an indirect assessment measure to gauge student satisfaction with the program. Also, we have developed several support systems to provide advisement and tutoring. We intend to assess these as well.

Conclusions What did you discover about student learning in your program?

Direct Assessment

Although the evidence is not conclusive, indications are that students who do not at least meet expectations for one learning outcome are likely not to meet expectations for another learning outcome. Those who meet or exceed expectations for one learning outcome appear to do so for most other outcomes. However, one major theme that was evident throughout the assessment is the lack of computer programming skills in a large percentage of our students. For example, in the Advanced Data Structures course that introduces advanced C++ programming techniques, 50% of the students did not meet expectations. Furthermore, it was evident that the large majority of the students lacked the basic C++ knowledge from the prerequisite courses (CSCI 271 and 272).

Therefore, policies must be developed to ensure all students moving forward in the major obtain the required competencies in a course and are prepared for subsequent courses. Development and implementation of such policies is timely, given the launch of the revised major that requires significant programming skills on behalf of the students.

Indirect Assessment

Enrollment trend data from the Office of Institutional Research indicate that during the last five years the number of declared Computer Science majors has increased by about 72%. This remarkable increase indicates growing interest in the Computer Science major, which is most likely due to a dramatically improved information technology job market. We expect these increasing enrollment trends to continue at a higher rate, given the need for cyber security professionals and the launch of the new Computer Science and Information Security major.

The gain in enrollment in the CSIS program reflects national trends. Data from the Office of Institutional Research show that between 1998–2001 (the dot com boom period), the number of declared CSIS majors averaged over 500 annually. With the collapse of the dot com economy and job market, came dramatic declines in enrollment, not only in our program but throughout the nation. (See the article http://chronicle.com/blogs/data/2014/06/23/is-there-a-crisis-in-computer-science-education/.) This report also shows that enrollments have begun to rise again nationwide during the past three years, a trend reflected in recent CSIS enrollments.

Actions Taken

What action decisions did you make based on your data and conclusions? (Plan actions to take effect in the following semester or sooner if practical.)

Actions To Be Taken and By Whom	Timeframe for implementation and intermediate steps
Continue to assess gateway, midlevel and capstone offerings. Incorporate guidelines from professional agencies into the assessment process. Computer Science and Information Security Coordinator, and faculty assigned to relevant courses.	Fall 2015, Spring 2016
Assess new capstone offering, CSCI 404. Instructor assigned to the course.	Spring 2016
Devise polices to ensure students do not progress in the major unless they show competency in entry- and mid- level courses (e.g., a grade of at least B in each of CSCI 271 and 272). Computer Science and Information Security Coordinator, and computer science faculty on the Dept. Curriculum Committee.	Fall 2015
Strengthen alignment of curricula and projects in CSCI 271, CSCI 272, and CSCI 373, which provide fundamental computing skills used throughout the program. Computer Science and Information Security Coordinator, and computer science faculty on the Dept. Curriculum Committee.	Fall 2015

Were last year's actions implemented as planned? Please explain.

Assessment last year allowed us to identify critical courses in the major where students must achieve the learning outcomes if they are to be successful in the program. We have begun to focus assessment on these courses and will continue to do so. We continue to assess the critical entry-level gateway course CSCI 272 − a prerequisite for all core level 300 courses that follow. In addition, we will continue to assess key mid-level core courses such as CSCI 373 and 377. These courses, typically offered fall and spring, respectively, provide fundamental knowledge and skills used throughout the major. In addition, student success in these offerings depends heavily on mastery of programming skills in the CSCI 272 course. We will continue to evaluate a capstone course at least once per academic year. The capstone requires students to apply a range of skills and knowledge acquired throughout the program.

Nevertheless, we did not assess the second capstone course (CSCI 404) this year. The reason is that most students taking the course do not have the necessary computer security background yet, as they are former CIS majors that switched to the new CSIS major. We plan to assess CSCI 404 next year, when students will be more prepared.

Regarding the alignment of the curricula in CSCI 271, 272, and 373, the Departmental Curriculum Committee has agreed to set up individual working groups for these courses. The groups will provide strict guidelines on the course curricula, exams, and programming projects that should be followed by all instructors. The goal is to ensure that students gain the necessary computer programming skills that are essential in the subsequent courses in the major. In addition, we plan to enforce a new requirement that students should maintain a grade of B or better in both CSCI 271 and 272 in order to complete our major.

Assessment data and conclusions were discussed in a Department or Program meeting on June xxx, 2015

Committee Members: xxxxxxx

Attachments: Please attach rubrics used and samples of student work at each performance level within the rubric.

Please find assessment reports for each course attached. Rubrics are included in assessment reports. Samples of student work also are attached.