

**CUNY Baruch College
Zicklin School of Business
Department of Statistics & Computer Information Systems**

CAP - CS Access Plan

Prepared by Professor Linda W. Friedman

This document is essentially a preliminary-stage strategic plan for increasing the participation of women in undergraduate Computer Information Systems programs and courses at Baruch College. Most of the strategies herein can probably be classified into the following categories:

- *Focus* – Our students love what they do but they, for the most part, are career-oriented. They want to know that there are jobs at the end of their studies and that they will be able to support themselves and their families on their earnings. In addition, they want work that is interesting and engaging.
- *Meaningful work* – Millennials in general, and women in particular, consider it of critical importance to do work that benefits society. They want to feel that what they do will make a difference in the short and long term.
- *Belongingness* – All students and women in particular, respond well to strategies designed to foster a sense of belonging – to a class, to a program, to a department, to a club, to a discipline. One theory regarding the dearth of women in technology is that they somehow don't feel that this field is for them, that it doesn't match with the way they self-identify.
- *Fun* – Coding is, or should be, fun. You can't do better than a field that is fun AND has jobs! Some of the strategies included here are designed to show more of the fun side of learning to code without degrading the learning process.

Background – Computer Information Systems is not Computer Science

At Baruch College, approximately 80-85% of undergraduate students major in a business discipline and earn a BBA degree. We are probably the largest undergraduate business school in the US. Once students are admitted to Baruch, if they wish to gain entrance to the school of business, they must complete eight pre-business courses with an average GPA of 2.25. BBA students have a required 30-credit business core, as well as the same "Pathways" requirements as all CUNY students. Thus, by the time students can enroll in classes in a major in the school of business, they are already juniors. Therefore, we do not have a lot of opportunities to begin working with students early in their college careers or even to recruit them to the CIS major until they are already well along in their studies. Not a lot of opportunities but, there are some, as we will see.

Gender Diversity in the CIS major at Baruch College

The graduation data below, from 2013-2014, shows that Baruch College holds its own within CUNY with regard to women the CIS major. However, while the data shows that the percent of women graduated is perhaps not as abysmal as some other schools (yes, faint praise, that), we are not where we could be.

Baccalaureate Degrees Awarded 2013-2014 Computer & Information Sciences (and related) in CUNY	Total	Men	Women	%Women
Bernard M Baruch College	126	93	33	26%

Brooklyn College	82	65	17	21%
College of Staten Island	50	43	7	14%
City College	29	25	4	14%
Hunter College	53	43	10	19%
John Jay College of Criminal Justice	27	19	8	30%
Lehman College	73	56	17	23%
Medgar Evers College	19	12	7	37%
New York City College of Technology	170	148	22	13%
Queens College	83	77	6	7%
York College	35	22	13	37%

Source: National Center for Education Statistics Integrated Postsecondary Educational Data System, data downloaded by Lecia Barker (WITNY), July 13, 2016

We need to start looking at this sort of data seriously. Recently, the Office of Institutional Research at Baruch College provided the data below related to gender diversity in three courses:

- CIS 3100 (Object Oriented Programming I), which is basically a CS1 course, is the first of a two-semester sequence in programming (in C++) required for majors (BBA/CIS) and for some business minors.
- CIS 2200 (Introduction to Information Systems and Technologies) is one of 8 pre-business courses required for entry into the School of Business.
- MTH 3300 (Introductory Computer Concepts) is a non-Business CS1 course in C++ (4 credits) taken by students majoring or minoring in Mathematics; these students are not in the Business School.

Term	Subject	Course	Gender						Total	
			Female		Male		Unknown		#	%
			#	%	#	%	#	%	#	%
Sum 2015	CIS	2200	101	47%	114	53%			215	100%
	CIS	3100	9	27%	24	73%			33	100%
	MTH	3300	10	45%	12	55%			22	100%
Fall 2015	CIS	2200	720	46%	850	54%	3	0%	1,573	100%
	CIS	3100	54	39%	86	61%			140	100%
	MTH	3300	35	43%	46	56%	1	1%	82	100%
Spring 2016	CIS	2200	633	47%	706	53%	3	0%	1,342	100%
	CIS	3100	36	26%	101	73%	1	1%	138	100%
	MTH	3300	39	49%	39	49%	1	1%	79	100%
Totals Over Sum 2015 - Spring 2016	CIS	2200	1454	46%	1670	53%	6	0%	3,130	100%
	CIS	3100	99	32%	211	68%	1	0%	311	100%
	MTH	3300	84	46%	97	53%	2	1%	183	100%

In the time period of this data, 3,130 students were enrolled in CIS 2200, a BBA core course required of all Business majors. Of these, females accounted for 46% and males 53%. During the same time period, 311 students were enrolled in CIS 3100 with 32% female and 68% male. It would have been nice to have

this data in sequential years rather than in the same year, and data collection and analysis will be part of the upcoming initiative. Still, it seems likely from this rough data that women major in CIS to less of a degree than other Business majors; there is no good reason for this disparity and we would like to address it. Note that not all students in CIS 200 will go on to major in Business as this is one of the pre-Business courses that students take before they are allowed into the Business programs. Also, note that CIS 3100 students may be CIS majors, CIS minors, students in other majors who use it as a free elective, and a smattering of others.

The Course: CIS 3100, Introduction to Object Oriented Programming I

As a central initiative in our goal to increase participation of women in CIS at Baruch College, we first focus on the introductory programming course, CIS 3300, Introduction to Object-Oriented Programming I. This is the first course of a two-semester sequence in OOP. It is probably comparable to many if not most CS1 courses in computer science programs. One difference is that we do not offer a 3 (or 4) course sequence in programming. Thus the first course goes all the way to programming with objects and classes; the second course (CIS 4100) covers more advanced topics related to programming “in the large” as well as some data structures topics. All CIS majors are required to take this course.

Course goals. The following are from the department’s CIS website:

Upon successful completion of this course, students will be able to:

- Describe the form and structure of the C++ programming language.
- Follow the steps in the program development process.
- Use common methods and algorithms used in computer problem solving and be able to express those algorithms in the C++ language.
- Develop solutions to a variety of programming problems using the C++ programming language.

Human goals: Basically, we want this course to, firstly, give students a good grounding in programming. In a nutshell, to learn to program in C++, including simple programs using objects. In a business program, students will not have the option of taking a large number of technical courses and yet, many of them will get jobs in software development, testing, etc. Every course counts. We can’t depend on them finally “getting it” down the road. From a purely procedural point of view, this course must prepare students for the second course in the sequence, so that there is not a lot of “catch up” work to do in the following semester or whenever the student takes the second course. Some students will not take a second course in programming; this may be their only experience with computer programming. Additional goals would include. Perhaps more important than anything, and hard to include in a list of sedate course goals, we really want students to get excited about coding.

Toolsets used. This course uses C++ in an IDE, such as MS Visual Studio. (The department also offers other courses that teach programming in Java and Python.)

Content. All course materials are available to students, instructors, and the public at large on a Google site at <https://sites.google.com/site/proffriedmanplusplus/> The course is offered over 14 weeks (plus 2 hour final exam), topics organized into six modules as follows:

- Module 1. Introduction to Programming in C++. Course Overview, Introductory Concepts, Hardware, Software, the Programming Process, Types of programming languages, Integrated

Programming environment. Introduction to C++ programming: identifiers, operators, types, input and output.

- Module 2. Controlling execution: Expressions. Formatting output. Debugging. Decisions. Scope. Loops. Counters. Nested loops. Using files.
- Module 3: What makes a good program? Program structures, data structures, stepwise refinement.
- Module 4: Program design. Defining and using Functions

Review / Exam #1.

- Module 5: Data design. Using Arrays (1-dimensional), Strings. Structures (records). File I/O.

Review / Exam #2.

- Module 6: Pulling it all together: Objects and Classes

Review for Final Exam

Without really altering the course content or the amount of programming students learn, this course can be delivered with some modifications to help us achieve the goals noted above. Some new ideas and strategies can be implemented in this first programming course. As an instructor, I can incorporate these into my classes right away. As course coordinator, I can encourage the other instructors to do the same. For example, the following:

Programming Assignments. Keep the same number of programming projects as we currently have, about 10. The work that students do on these projects are incorporated into exams to encourage honest work.

- Group Work. Currently projects are individual work. I'd like to change several to group assignments (2-3 students), and facilitate diverse groups as far as possible.
- Realism. Examine the programming projects to try to make them more realistic. Create one assignment that uses real data.

Make the Coursework More Fun. Programming *is* fun and we want to transmit this excitement to our students. Some ways to do this:

- Hour of Code. In the first week of class, students will meet in a lab and run the Hour of Code games on code.org.
- Use more visual examples, like the "birthday" data to illustrate use of a 2-dimensional table.
- Text adventure game.
- Change one or more programming project to apply to gaming.
- Scratch programming. Can set up online lectures with lab assignments.

Other teaching practices to promote

- Meet Me Assignment. Students get 1-2 pts for coming to talk to the instructor. This helps to promote belonging, and can sometimes solve problems before they come up. If nothing else, once the student visits the instructor in his/her office for the points, the student will then know where the instructor may be found, and realize that the instructor welcomes the visit.
- Computers in Service to the World. A two-page paper citing 3 examples of ways programming has helped the world, i.e., served a higher societal purpose. (Saving lives would be nice.) This assignment will get students actively involved in learning about the

field as one which is meaningful. This is expected to help with retention and word-of-mouth recruitment, especially of women.

- **Name Cards.** To increase class participation, call on students to participate / answer questions by use of name cards in a deck. Each card has one student name; the cards are in random order. Call on student, place that card at the bottom of the deck.
- **Study Groups.** I'd like to encourage and facilitate students to form study groups, preferably diverse – or at least mixed-gender – groups.

Some possible strategies would support teaching and learning in this course, but are not in the control of the faculty and so are more difficult to bring to fruition. For example:

- **TAs in open labs during set scheduled hours.** Ideally, we want a diverse group of TAs, so that women and members of underrepresented groups will feel comfortable and fit it. As wonderful as this would be it requires resources, including a TA director, which we don't have right now. It stays on the list, however.
- **"Mobile labs."** For lab sessions, the instructor can suggest that students bring their personal laptops to class; many of them do this anyway. Currently, students who don't do this can borrow loaner laptops from the Library. Rather, a cart with 10-15 loaded laptops can be delivered to the classroom as needed for students who don't bring in their own.

Likes / Dislikes / Desires / Constraints:

Likes. The course gives a good, solid overview of programming. It can take a total "newbie" and turn her or him into a fairly proficient programmer by the end of the semester. Students work on small to (somewhat) mid-size programming projects.

Dislikes. The course requires that students do a lot of programming work on their own (or perhaps in pairs) in order to become "real" coders. This is not really a Dislike, more of a challenge. Students may work at differing speeds with some students "getting it" right away and some needing to go through the material in a more deliberate manner.

Desires. We need more labs, and lab assistants. We will not get them.

Constraints. Space is a limiting constraint. Money for such initiatives – for example, lab equipment, lab assistants, Hopper Conference registration and travel expenditures, departmental student advisors – has been a problem for a long time.

Recruitment

Recruiting within the CIS 3100 course:

As noted above, because of the process by which students are admitted to the business school and the CIS (BBA) major, recruiting students (both male and female) into the major is a challenge. By the time we see students in CIS 3100 they are likely to already be declared CIS majors; most are juniors in their first semester of coursework in the major. All this means that we have not given much thought to using CIS 3100 as part of a recruitment plan. However, upon reflection, there are opportunities for recruitment within this course in addition to those strategies we are considering that are external to this course.

Still there are plenty of strategies that we can use in this regard. There are already non-CIS majors taking CIS 3100. These include students taking the course as part of a CIS minor, students preparing for a graduate degree in financial engineering and other math majors, students taking this and other technical courses as electives, to give them breadth and add important skills to their resumes. With all the “chatter” currently heard about the benefits of studying technology, we can:

- Use this course to promote our major to students who presumably are already halfway convinced
- Promote this course to students in other majors either as a highly appropriate elective or as part of a minor in CIS.

Of course, if we are going to use this particular course as an attractive recruitment tool, we will have to make sure that it is interesting, useful, accessible and – oh, OK, how about – fun! Part of this strategy would include making the abovenamed changes noted above (in the Course section), including a better name for the course. (Object oriented programming? Really? Do we teach any non-OOP coding?)

Strategies external to the CIS 3100 curriculum:

The overarching message to get out to potential students has four parts, as discussed earlier:

- Focus – There are jobs in technology; they pay well, and the work is interesting.
- Meaningful work – Work in technology benefits society; it makes a difference in the short and long term.
- Belongingness – We want to create and maintain a sense of belonging among our students; ultimately, this will be one of our biggest selling points.
- Fun – of course.

And, yes, CIS is – or can be – all these things, whether we are talking about working in software development or as part of an IT startup team; majoring in CIS; or using essential technology courses to provide the breadth for another major, such as accounting, finance, marketing, human resources, or even non-Business majors like psychology, economics, math.

We can and should communicate beyond the students we already have.

- Run an Information Session for undecided students.
- Man (woman) a table on Majors & Minors Day. Need large freestanding posterboard, informational cards, other materials.
- Prepare information (posters, brochures, cards) for the Advisement office. How about high school advisors? I think for that we have to work through the College.
- Can we recruit at Baruch’s Early Learning Center (i.e., a day care center)? Should we?
- Create a page on the department website that highlights successful grads. Bios, pics, video interviews. Will need support for this.
- How about creating an advisory group (of faculty, admins, industry experts) for WIT. Alternatively, this can be a “Diversity in CIS” committee.
- Use the pre-Business courses CIS 2200 (Intro to Information Systems and Technologies) and STA 2000 (Intro to Business Statistics) to promote our tech courses.
- Find out – what do students want? What speaks to them?
- Two female students go to (are invited by the instructor) sections of CIS 2200 and STA 2000 to promote majoring in CIS (and minoring and just taking a free elective). How to promote course to other majors? Examples of how CIS meshes with other disciplines and why learning programming is a good idea for Finance, Accounting, Marketing, Law, etc.

- Unlike other programs, we do not have a CS0-type of course, which is sometimes considered a recruitment tool – a way of demonstrating to students early in their careers the advantages of working in technology. Should we create one? A gen ed course? It might be worth looking into.

Retention

Data, data, data. Just like we need to start collecting data for our recruitment efforts, we need this even more so for our retention strategies. Currently, we have no idea at all what our retention rates are like.

One possible retention strategy is to embed student advisors in the Department. Currently, students use the college-wide advisors until they are admitted to the Business School (I think). Then there are Business-specific advisors in the Dean’s office; those are much, much better than the college-wide advisors.

- Some community colleges have advisors (HEOs) in the department itself. This would be very useful.
- Alternatively, we can try to organize and publicize a schedule for faculty who agree to make themselves available for student advisement.

Many retention strategies focus on the notion of “belongingness.” For example:

- Send students to national conferences like the Grace Hopper Celebration of Women in Computing
- Space for students to hangout – a CIS student lounge – in the department, close to faculty. Make it a welcoming space.
- Joint events with the student club(s). This also should help with recruitment, e.g., workshops, panels.

Reflections

This is a major step for our department and, of course, an ongoing process. We need to build a core group of dedicated faculty to lead this effort, while maintaining documentation that is updated regularly to provide for institutional memory.

Next Steps.

- We are working towards membership in NCWIT and would like to participate more in WITNY events and activities, and other national organizations like ACM-W and Girls Who Code.
- We have a close relationship with a newly organized student club at Baruch, FemCode, for men and women who support women in technology.
- We are planning a meeting for interested faculty early in the semester to brief them on the CUNY/WITNY workshop and solicit doable solutions to implement some of the strategies in this document.

CUNY faculty working together. We would like to build and participate in some CUNY-wide initiatives such as:

- A CUNY Women in Technology Conference
- A CUNY Undergraduate Research in Computer Science Conference
- A repository of CS0 and CS1 course materials across CUNY