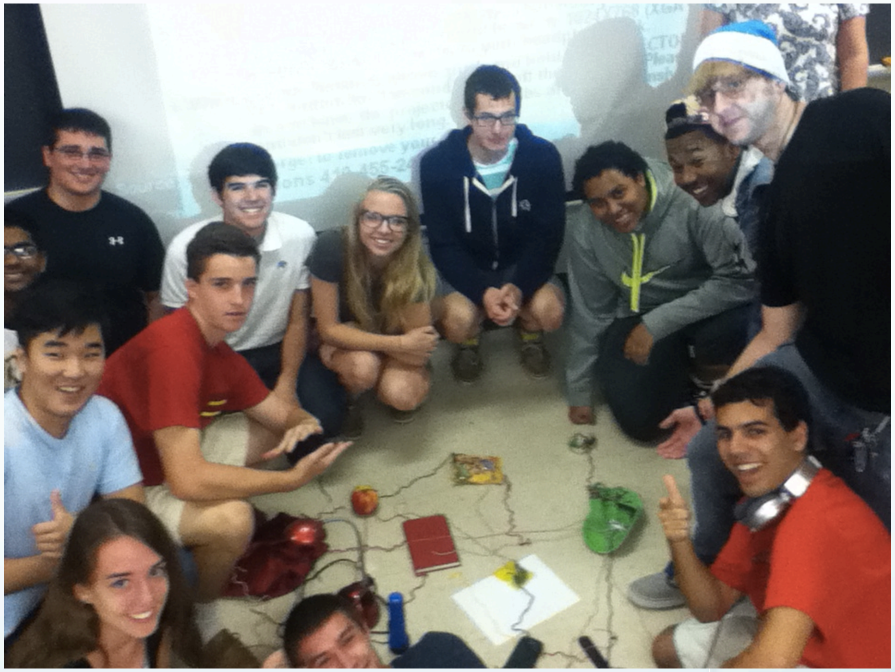
COMP 101: Designing a First-Year Experience for Computing Majors

Project Findings

Penny Rheingans
Carolyn Seaman
Marie desJardins
Susan Martin

May 9, 2014



Overview

- Course content
- Implementation logistics
- Peer mentoring
- Evaluation and assessment
- Preliminary results
- Challenges and lessons learned

Course Content

Four Curricular Arcs

- Overview of the discipline
- Key technical skills
- Group design and implementation experience
- Academic and professional skills

Overview of the Discipline

- **Big Ideas:** computational thinking, algorithmic problem solving, abstraction, history, theory of computation
- **Data:** representation, design and modularization, data structures, big data, visualization
- **Hardware and Systems:** beginnings of design, computer architecture, operating systems, networks
- **People:** analysis and requirements, usability, HCI and accessibility, social and ethical implications
- **Applications:** graphics and games, intelligence, security

Key Technical Skills

- Algorithmic design
- Some programming
- Data analysis

Group Design Experience

- **Semester Game:**
  - Simulation of 15-week semester
  - Players make choices about how to allocate time
  - Game calculates outcomes of those choices
  - User interaction and presentation
- **Phases:**
  - Design
  - First demo
  - Demo evaluation
  - Poster
  - Presentation

Professional Development

- Understanding learning styles
- Academic success skills: time management, test taking, using academic resources
- Working effectively in teams
- Giving and receiving feedback
- Giving clear and effective presentations
- Understanding degree requirements and career planning
- Professional networking

Implementation Logistics

Course Staff

- **Technical Content Instructors:** Penny Rheingans (1), Marie desJardins (2), Carolyn Seaman (2)
- **Professional Development Instructor:** Susan Martin
- **Teaching Assistants:** Emily Scheerer (1,2), Alex Pulianas (1,2), Bhuvana Bellala (2), Amanda Mancuso (2)
- **Peer Mentors:** Marie Wagner (1), Tiffany Ernst (1), Clayonna Wheat (1), Max Weinberg (1,2), Mark Cirincione (2), Gloria Diederich (2), Catherine Liou (2), Nicole Dawson (2), Brandon Walsh (2), Austin Cole (2)
- **Assessment Intern**: Ashley Reed

Team-Based Learning

- Used elements of Michaelson's Team-Based Learning (TBL)
- Students assigned to teams
  - Yr 1: randomly (almost) on first day of class
  - Yr 2: randomly on first day, then again later intentionally
- Activities with team
  - Team quizzes
  - In-class exercises and discussions
  - Data analysis assignment
  - Project
- Grades: 35% team components, 10% peer assessments, 30% individual assignments, 25% tests and quizzes

Assignments

- Journal entries: StrengthsQuest reflection, academic and study habits questionnaire, teamwork reflection, study habits reflection, final course reflection and survey
- Programming assignments in Processing
- Resume and cover letter
- Presentation of data about enrollments, progression, and diversity of COEIT majors (year 1) or time management behaviors (year 2)
- Team project with design, implementation, evaluation, and presentation phases

Peer Mentoring

Peer Mentoring

- Integral course element: peer mentors (sophomores) and TAs (juniors or seniors)
  - Based on Peer-Led Team Learning model
- Research on peer interactions has shown:
  - Increased achievement and persistence
  - Cognitive growth
  - Development of leadership

Peer Mentors

- Four (4 hour/week) peer mentors per class of 40 students
- **Responsibilities:**
  - Attend regular discussion sections (two peer mentors per 20-student discussion session)
  - Assist with class activities, discussions
  - Coach project teams (meetings outside of class to handle interaction issues and challenges)
- **Selection criteria:**
  - (Ideally) recent student in COMP 101
  - Introspection and self-awareness about major selection, college success, peer interactions)

Undergraduate TAs

- To w(10 hour/week) TAs per class of 40 students
- **Responsibilities:**
  - Attend regular class meetings
  - Assist with class activities, discussions
  - Prepare and lead one class session
  - Grade student assignments
- **Selected for:**
  - Ability and interest in teaching and mentoring younger students
  - Technical strength and confidence

Peer Mentoring Logistics

- **Retreat/orientation prior to semester**
  - Set expectations
  - Familiarize staff with course structure
  - Establish schedule
  - Build community among course staff
- **Weekly meetings during semester**
  - "All hands on deck": instructors, PIs, TAs, peer mentors, research assistant
  - Discuss successes and challenges of previous week
  - Discuss expectations, tasks and concerns for upcoming week
  - Touch on longer-term planning and tasks

Evaluation and Assessment

Project Goals

- NSF funding requires extensive plans for **goal-based assessment**
- Project goals:
  - Increase retention within computing
  - Increase graduation rate in the computing majors.
  - Increase the overall success (academic performance) of participating students.

Learning Goals

- Increase understanding of the discipline, in terms of different majors and careers.
- Clarify students' personal interests and motivations about their choice of major and career.
- Increase confidence, self-efficacy, and community.
- Expose students to, and allow them to practice, key design and development skills.
- Teach skills in problem solving, algorithmic analysis, and computational thinking.
- Help students learn how to study effectively and how to access campus academic resources.

Methods

- Control group
  - Freshmen in IS 101 or CMSC 104 (the courses that our students would take if there were no COMP 101)
  - Had to consent to participate in research study
- Data collection
  - Student records
  - Focus groups – experimental and control, three times during the semester
  - Pre and post survey
  - Exit interviews – experimental and control, once each semester
  - Journal entries
- Data analysis – both qualitative and quantitative

Evaluation Challenges

- Participation
- Small population
- Long horizon
- Self-reporting

Preliminary Results

Major Switching

- Major switching
  - We “lost” about 18% of the control group to non-computing majors within a semester of taking the control course
  - For the experimental course, this figure was only 10%
  - Overall, there was less major switching in the experimental group (20%) than in the control group (45%)
  - Very little of the major switching appears to be due to low grades
  - Much higher proportion of women switching to non-computing majors in the control group than in the experimental group

Future Behaviors

- Most students in both groups who continued in a computing major also continued to take computing courses
- Students in the experimental group did slightly worse (average 2.68) than students in the control group (average 3.05) in computing courses in the following two semesters
- Grade in experimental course not as good a predictor of success in future computing courses as grade in control courses
- No trends in terms of gender or ethnicity

Introspection

- Qualitative, subjective observations from focus groups and exit interviews
  - Shed light on learning goals
- Experimental group students were thinking more about, and asking more in-depth questions about, computing majors and careers
- Experimental group students learned some of the complexities of team work
  - Reported knowing LESS than they did at the beginning of the semester about effective study groups
  - Experimental group students were able to describe and reflect on their teamwork experiences and could apply their experience to future coursework

Programming and Computing

- Programming was a source of anxiety in the experimental group:
  - eroded confidence
  - felt that they were not “getting it”
  - thought that other students were much further ahead
  - led some students to question their suitability for a computing major
  - in retrospect, felt that the programming skills they learned in the experimental course were very helpful when they took the follow-on programming course in their major
- Experimental group understood computing concepts such as algorithmic problem solving and computational thinking in the abstract, but expressed difficulty in applying these concepts to the work they were doing in the course, or to any other real problems.

Challenges and Lessons Learned

Clear Successes

- Peer-to-peer interactions with peer mentors and undergraduate TAs
- Group quizzes as a learning mechanism
- Active classroom engagement
- Team project as a learning experience
  - Perceived as very difficult
  - Students were *very* proud of their results (confidence building through challenging tasks as a team)

Challenge: Role of Programming

- Mix of programming experience and confidence among students
- Relatively little time spent during class teaching and practicing programming skills
- As a result:
  - Experienced students sometimes took over the team
  - Less experienced students sometimes felt helpless and unable to contribute
- Possible solutions:
  - Spend more time teaching programming?
  - Separate students with different experience levels?
  - Downplay programming and emphasize other skills in the project?
  - Don't do programming? (what would the project and technical skills include in that case?)

Challenge:
Mix of Students

- Different majors
- Different level of programming experience
- Different mathematical sophistication/ability
- Different technical depth
- Different goals

- ...maybe *too* different?
  - Experienced/technical students tend to intimidate less experienced/less technical students
  - Many institutions have been trying to separate students with different experience levels because of this challenge

Challenge: Classroom Environment

- Year 1:
  - terrible environment – computer lab with fixed desks, monitors blocking sightlines; complete inability for teams to work together effectively
- Year 2:
  - Section 1: better environment – regular classroom with movable desks; teams could move into groups but not easily
  - Section 2: bad environment – lecture-style classroom with long fixed desks, difficult for teams to work together
- Year 3:
  - ACTIVE Center: new classroom – ideal environment (movable desk groupings, movable whiteboards for every team, smart projector and flat-panel displays for sharing groupwork)

Challenge: Textbook/Readings

- Year 1: collection of readings – not very successful (students didn't see the purpose, no cohesion)
- Year 2: Robert St. Amant, "Computing for Ordinary Mortals" – mixed reactions
  - One instructor liked it; one didn't
  - Mix of very non-technical and rather technical discussion
  - Many students didn't like the style of the book and couldn't relate to it
  - Hard to identify concrete knowledge from the book that students should know / be quizzed on
- Year 3: ???

Challenge: Relationship to Curriculum

- Originally could substitute for two courses:
  - CMSC 104 – taken by CS and CE students without prior programming experience or necessary math placement for CMSC 201
  - IS 101 – recommended (IS) or required (BTA)
- Things have changed...
  - Students without prior programming experience can go directly to CMSC 201
  - IS 101 is being significantly redesigned and is still in flux
- Not clear how this course fits into the curriculum, or which students would/should take it