Overview
Our new first-year course for students in four computing majors (CS, CE, IS, and Business Technology Administration) is designed to increase retention, completion, and success among students, especially women and underrepresented groups. The approach synthesizes elements from successful first-year engineering courses, traditional introductory computing courses, general first-year seminars, and the new AP CS Principles course. Course content is woven into four curricular arcs designed to provide an overview of the discipline, build technical skills, provide a group design experience, and strengthen professional skills. Learning infrastructure includes collaborations among experienced computing faculty, staff members with student affairs experience, peer mentors, and undergraduate peer teachers. We offered a pilot section in Fall ’12 and will offer two revised sections in Fall ’13. We plan regional workshops on the topic of introductory computing for the summers of ’14 and ’15.

Core Learning Goals
This course includes specific elements designed to achieve core learning goals:

1. Increase understanding of the discipline, in terms of different majors and careers.
2. clarify students’ personal interests and motivations about their choice of major and career.
3. Increase confidence, self-efficacy, and community.
4. expose students to, and let them practice, design and development skills.
5. strengthen writing, presentation, and teaching skills.
6. teach skills in problem solving, algorithmic analysis, and computational thinking.
7. help students learn how to study effectively and how to access campus academic resources.

Course Elements (pilot offering)
Course Staff: one faculty member (CS), one staff member with a background in Student Affairs, two junior teaching fellows (CS, CE), four sophomore peer mentors (CS, IS), one assessment intern
Survey of Computing Concepts and Areas: Big Ideas, People, Data, Hardware and Systems, Intelligence
Integrated Professional Development Track: learning styles, time management, test taking, teamwork, academic resources, feedback, presentation skills, degree requirements, career planning, networking
Assignments: five journal entries; programming assignment in Processing; Matlab exercise; resume and cover letter; presentation of data about enrollments, progression, and diversity of CS&E majors; project with design, implementation, evaluation, and presentation phases
Assessments: pre- and post-survey of student attitudes about the computing field, confidence about ability to succeed, and aspects of the course; analysis of written reflection about career goals; focus groups of students from experimental section and comparison sections; longitudinal tracking of student outcomes.

Team-Based Learning
We introduced elements of Michaelson’s Team-Based Learning approach in order to increase engagement, reinforce connections among students, and enhance learning. The team-based learning approach replaces lectures with in-class team activities. Students were randomly assigned to teams of about five on the first day of class. The team organization was the basis for team quizzes, in-class exercises and discussions, the data presentation assignment, and a multi-phase term project. Students completed peer evaluations of their teammates at midterm and the end of the course. Team-based components made up 35% of students’ grades, while peer evaluations made up another 10%. Students found interactions with their teams to be a source of enjoyment, support, learning, and frustration.

Term Project
Each team used Processing to create a game based on a college scenario where players try to win by creating the most effective balance of grades, happiness, and wealth. Ideally, one would want to maximize all three, but, more realistically, the three objectives compete with one another. The game simulates a fifteen-week semester. During each week, the player makes choices about how to spend their time. Two choices are made at the beginning of the semester and remain the same through the whole semester: number of credits of classes and number of hours worked at a job. Eight more choices are made each week: hours spent attending class, percentage of class spent actively participating (and not surfing, texting, sleeping, etc.), hours spent studying and working on assignments, hours spent participating in study groups, hours spent using academic resources (professor/TA office hours, help center, LRC, etc.), hours spent on taking care of self (sleeping, eating, exercising, etc.), hours spent on solitary leisure activities, and hours spent on nonacademic social activities (time with family, church, clubs, etc.). Project phases were a design document, a demo and sales pitch, an analysis of design and functionality, creation of a poster, and presentation at a public poster session.

Conclusions
We experimented with a number of things. Some were successful (The Good), some weren’t (The Bad), and some were especially disappointing (The Ugly).

The Good
Undergraduate teaching fellows and peer mentors Group quizzes as a learning mechanism Term project producing impressive results

The Bad
Dependence on a collection of papers instead of a textbook Insufficient explicit programming instruction Singleton assignments/exams without opportunity to build confidence

The Ugly
Student ambivalence and/or dissatisfaction with concept Difficulty comparing content mastery with control sections Completely horrible and inappropriate classroom