Reflections on an Introductory Computational Mathematics Course

Jillian Cannons

California State Polytechnic University, Pomona

Outline

1. Course Motivation
2. Course Structure and Logistics
3. Math Content
4. Reflections
5. Conclusions

Course Motivation

- Goal is to introduce *all* math and stats majors to computational mathematics
- Benefits include
  - building problem solving skills
  - building programming skills for use in
    - making math fun and approachable for young students
    - upper division applied / stats courses
    - industry
- Traditional introductory programming courses are designed for computer science students, not math and stats students
- Introduced MAT 2010/L in Fall 2018

Course Structure

- Weekly contact hours: 2 hours lecture, 3 hours lab
- First CPP math and stats class with a lab
- Lab facilitates CPP's "Learn by Doing" philosophy
- Scheduled for two days a week, e.g., Wednesdays and Fridays
  - Lecture 2:00pm - 2:50pm
  - Lab 3:00pm - 4:15pm
- Lecture can occur the first meeting of the week for the needed time (variable but often about 1.5 hours) then flow immediately into lab time

Classroom

- Requires a room that can easily convert between a standard whiteboard lecture room and a computer lab
- Renovated a lab with a standard setup to use Smartdesks flipITlift desks

www.smartdesks.com

Assignment Feedback and Grading

- Original Approach
  - Manually reading every line of code submitted for each weekly assignment is very time consuming
  - Wrote scripts to run student code with various input and check the output
  - Sample input and output is provided for each question on each weekly assignment
  - Students were expected to devise and verify other test cases
  - Students were disheartened when their code failed cases tested in my scripts
- Refined Approach
  - Desired a means for the students to test their code using my scripts before final submission
  - Gradescope facilitates an autograder that uses my scripts to provide near-instant feedback
  - Significant increase in student engagement as they were motivated to find and fix their mistakes
  - Manual grading is performed for good programming practices

Math Content

- Lectures are organized weekly by programming content with math interspersed
- Math content is selected to coincide with weekly programming content
- Sample topics and problems
  - Stats: Min / max / mean / standard deviation, Markov processes
  - Probabilistic Algorithms: Primality testing, Monte Carlo methods
  - Number Theory: Factors, Euclidean algorithm, cryptography
  - Combinatorics: Balls and urns
  - Linear Algebra: Matrix multiplication, eigenvalue and eigenvectors
  - Graph Theory: Representations, breadth / depth first search
  - Numerical Methods: Numerical integration / differentiation, rootfinding
  - Games: Tic-tac-toe, card games

Reflections

- Introduce programming content very slowly
- Keep lectures short and interactive
- Students struggle with loops, largely because they permit more in-depth problems
- Students struggle with problem solving
- More students are excited about programming and ask about follow-up courses

Conclusions

- Developed an introductory course in computational mathematics with a lecture and lab
- Strive to incorporate a broad range of math and stats topics
- The hands-on aspect of the lab, coupled with the classroom and grading setup, has been very successful
- Students now want to learn more programming!

Thank You!

Questions?