Renaissance Computing @ UNL
CURRICULUM PLANNING WORKSHOP
Computational Thinking (Wing 2006)

- Computer science is the study of computation—what can be computed and how to compute it.
- But what is computational thinking?
  - Using abstraction and decomposition when attacking a large complex task or designing a large complex system.
  - Using heuristic reasoning to discover a solution.
  - Using massive amounts of data to speed up computation.
  - ...
Computational thinking has the following characteristics:

- Conceptualizing, not programming
- Fundamental, not rote skill
- A way that humans, not computers, think
- Complements and combines mathematical and engineering thinking
- Ideas, not just artifacts
- For everyone, everywhere
To teach computational thinking to everyone on campus may require different approaches than those we use when we can assume our students want to become computing professionals.

- What do non-computing students understand about computing?
- What will they find challenging?
- What kinds of tools can make computational thinking most easily accessible to them?
- How should we organize and structure CS classes?
To make “computational thinking” accessible to students across the entire campus, we need to understand how to teach computing better:

- How humans come to understand computing and how to improve that understanding?
- How to draw on a variety of disciplines to make computing education better?
- How do we teach every student programming and the theory of computation in a way that makes sense to them for their discipline?
Renaissance Computing @ UNL

- Funded by the NSF CISE Pathways to Revitalized Undergraduate Computing Education (CPATH) Program
- Funding Period: August 01, 2008 – July 31, 2010
- Grant type: Concept Development and Planning ($100K)
- Team Members: Leen-Kiat Soh, Ashok Samal, Stephen Scott, Stephen Ramsay, Etsuko Moriyama, George Meyer, Brian Moore, William Thomas, Duane Shell, and Usha Chandra
> Vision

- We envision a **campus-wide undergraduate program** in which **students are prepared for the challenges of any one computational domain** by virtue of their exposure to multiple domains in which **computational thinking plays a prominent role**.
The Basic Framework

• Our framework
  • Emphasizes interdisciplinary course contents
  • Considers undergraduate computing education for CS majors, CS minors, and non-majors (flexibility)
  • Addresses introductory, depth, and capstone courses (stewardship)
  • Incorporates technology, problem-based learning, collaborative learning activities (pedagogy)
Renaissance Computing CS Courses

• The CS1 “Funnel” (required of all CS majors/minors/non-CS majors)
  • E.g., CS1-Engineering may use C as the programming language with lab assignments in simulation; CS1-Arts may use Python as the programming language with lectures in HCl; etc.
  • All CS1 courses will contain the same basic core of CS topics (as identified in IEEE/ACM Computing Curricula 2001)

• CS2 (required of all CS majors/minors)
  • Data structures, searching/sorting, OO concepts (according to CC 2001)
• Depth Courses (tech electives for CS majors/minors/non-CS majors)
  • 400/800-level courses
  • Simulation? Computer Visualization? Autonomic Computing?
• Capstone Course (required for all CS majors/minors; tech elective for non-CS majors)
  • Project-based
Technology & Collaborative Learning

• Use of learning objects to supplement instruction
  •Expose CS1 students to contents in other funnel courses
  •Allow for more flexible course development and revision

• Use of online collaborative learning systems
  •Promote teamwork
  •Encourage interdisciplinary discussions among students and instructors
  •E.g., Wiki-type collaborative systems
Planned Activities

• Year 1: Planning and Development Phase
  • Today’s workshop for curriculum planning
  • Initiate and support campus-wide forums
  • Develop at least two CS1 courses (CS1-Sciences and CS1-Engineering) for pilot study
    – Develop several learning objects
  • Get collaborative learning system ready
Planned Activities, Cont’d …

• Year 2: **Pilot Study Phase**
  • Prepare educational research design and apply for IRB approval
  • Collect results and analyze
    – Knowledge and skills
    – Attitudes, motivation, self-efficacy
    – Course impact on students’ perception and self-regulation
    – Feedback on technical aspects of course implementation (from students and instructors)
  • Prepare and submit **Transformative Implementation** (TI-type) grant to NSF CPATH ($1 million)
For Today’s Workshop

- Breakout sessions into groups
  - A Renaissance Computing team member will lead
  - Several tablets for you to use
- Surveys
  - What you think are important topics for your students to know
- Breakout Worksheets
  - A set of 8 questions as discussion points
  - Other questions are welcome
- Breakout Session Report by group leaders
  - Q&A afterwards
For Today’s Workshop

• Breakout sessions into groups
  • A Renaissance Computing team member will lead
  • Several tablets for you to use
  • Surveys
    – What you think are important topics for your students to know
• Breakout Worksheets
  – A set of 8 questions as discussion points
  – Other questions are welcome
• Breakout session reports by group leaders
  • Q&A afterwards