

---

# CC2020 - a substantial undertaking

Simon, *University of Newcastle*

Project led by

Alison Clear, *New Zealand, for ACM*

Allen Parrish, *USA, for IEEE-CS*

---

---

# ACM/IEEE computing curriculums

- CC2005
  - the most recent overarching volume of undergraduate programs in computing
  - encompasses the (US-style) disciplines of CE, CS, IS, IT, and SE
  - sits above individual discipline-specific volumes
  - essentially compares the disciplines in terms of coarse-grained content, graduate outcomes, likely employment opportunities, etc
- CS2013
  - one of the more recent discipline-specific volumes
  - far more fine-grained in terms of curricular content

---

# CC2020

- CC2020 will be the next overarching volume
- It has the potential to redefine the constituent disciplines of computing
- The task force consists of 31 people from 14 countries
- Dominated by the US (17), but enough others to ensure some balance

---

# The initial survey

- The project began with a survey, mainly about how people have been using CC2005
- Thanks to ACDICT, Australia had the second-highest number of responses, 57
- (just behind the US, with 720)
- Analysis has barely begun
- But we thought it might be interesting to share some answers to the question “What do you think computing education will look like 10 years from now?”

---

# Elementary response counts

- Total survey responses: 1186
- Skipped this question: 736
- Effectively null answers: 29
- Answers saying no change: 15
- Answers simply saying it will be different: 7
  
- Leaving 399 responses imagining how computing education will be different 10 years from now
- On the remaining slides we'll take a look at some of the dominant themes
- (Some responses covered more than one theme)

---

# Changes of focus (88 responses)

- cybersecurity
- data science
- artificial intelligence
- cloud computing
- human-computer interface
- machine learning
- robotics
- virtual reality
- distributed computing
- internet of things
- mobile app programming
- parallel computing
- privacy, ethics
- problem solving
- quantum computing
- and many more

---

# Interdisciplinary (63 responses)

- CS+X
- embedded
- pervasive
- design your own degree
- educating far more non-majors
- In . . .
  - arts
  - business
  - criminal justice
  - data journalism
  - digital arts
  - engineering
  - health
  - human sciences
  - humanities
  - music
  - sciences

---

# Online (48 responses)

- More on-line or company sponsored sources of learning computing
- Virtual, no face to face classes or lectures
- Less reliance on direct lectures and more use of technology
- Increased reliance on online models and automatic evaluation with instant feedback
- Sadly I think that the trend is towards online self-paced learning
- Sadly mainly online, which is NOT what the industry wants but where Universities are heading for resource reasons

---

# Pedagogical alternatives (22 responses)

- Like a Design Studio - practice based, experiential
- More emphasis on online resources and flipped classrooms
- Virtualization, cloud resources and containers will make it easier for students to interact and use many heterogeneous HW environments without the setup pain
- Mostly taught in AI assisted learning environments, continually evaluating students and individualizing training
- The sharing economy will have an impact. We have taxi companies that do not own a single car; will we have schools without lecturers or students?

---

# Vocational (20 responses)

- Intensive courses rather than degrees and diplomas
- Micro certificate based, continuous updating
- More certification oriented, not requiring a college degree
- I am afraid that it will be mostly focused on training for specific technologies, without much scientific content or mathematical background
- Less theory and more hands on - certification driven
- On-line micro degree(s) that are more responsive to the immediate needs
- Most people will receive it from community colleges, high school, and for-profit bootcamps and not universities

---

# Responsive (19 responses)

- It will be more advanced and what we are teaching now will just be part of the history
- It is likely to be tied to short term fads and unlikely to have a long term view
- For undergraduates, less theory than we now teach and more oriented toward market driven needs
- More responsive to the demands of business and industries than focusing on learning fundamentals
- Whatever is number 1 on the TIOBE index at the time, in the misguided belief that students will get jobs right away
- No change except schools will offer the latest in languages, frameworks, shiny objects, and flying pigs

---

# Fragmented (16 responses)

- More distinction between CS, CE, Security, IT, IS, etc
- We seem to be splintering more and more (cyber security, data science, etc)
- Fractured, messy, still fighting about languages, growing
- More fragmented, continuing to move away from the needs of business and industry and focusing more on gaming and entertainment
- More splintered, less core curriculum, more different options, less consensus about what "everybody needs"
- More fragmented than ever, with core fundamentals increasingly abandoned in favor of the flavor of the month. Maybe I'm just feeling dystopian these days.

---

# Pervasive (16 responses)

- It will be a component of every other educational program as the world is becoming increasingly computerised. Programming for example will be expected of everyone, similar to the way writing is.
- Not only students but ordinary people will take computer education
- It should be pervasive in all disciplines - like Math or Grammar
- Ubiquitous and pervasive, due to the IoT
- I think it will be part of every student's education and the battles I have fought over the last 30 years of teaching will be won

---

# Schools (16 responses)

- More widely spread through schools / K12; not properly resourced; teaching highly variable
- There is a huge need for computing education in a variety of contexts from elementary school to life long learning
- Much altered by schools teaching more and students starting with much more knowledge, but still with lots to learn
- I am hopeful that higher ed can teach more in-depth material to a wider audience once our students have come through the much-expanded K-12 offerings that are coming into being now

---

# Diverse (12 responses)

- More diverse in topics and student origin
- We will continue to face diversity issues that will need to be addressed. Due to the rise of "bootcamps" and similar activities, we will be encountering a large number of people who think they understand what computing is, but have a very limited worldview.
- Broader, more diverse curricula based on student needs
- Without radical intervention, it will change much too slowly to accommodate the demands for more skilled individuals, including a larger pool of diverse students
- More diversification of computing disciplines, and more application of computing in all areas of life

---

# Broader picture (11 responses)

- I hope there will be more emphasis on themes than today and less emphasis on a catalog of details
- More high-level and more specialized, I expect; like we don't spend much time teaching school pupils what the pen ink is made of, but rather teach them how to write
- Less emphasis on lower-level (nuts and bolts) and more on solving problems with greater complexity
- Hopefully, will provide an integrated view of systems, hardware, and software
- Less focused on "Coding" which will have been pushed down into pre-college for most students. More focus on design and higher level aspects.

---

# Fundamentals (11 responses)

- I hope there will be a focus on fundamentals because these remain relatively constant
- The "basic fundamentals" I hope will still be there
- I hope a move to learning the concepts of computer science in a logical manner, not just random code snippets to learn Java or some other language
- The core of CS will not change massively
- Core theory is sound and should remain
- How do we teach computing for the things that haven't been invented yet? Remain focused on the essential fundamentals so that students can learn beyond their college career.

---

# Industry-based (11 responses)

- Possibly much more plugged into (and offered by) industry
- There will probably be more on-line or company sponsored sources of learning computing
- Everything should be hands-on and include a real-world example for a live company
- Alignment with industrial BoKs
- Take industry on board
- More collaboration between industry and academia
- More responsive to the demands of business and industries than focusing on learning fundamentals

---

# Specialist (10 responses)

- CS programs will further segment to specific areas of study (ie HPC, Big Data, Soft Eng)
- There will be more tracks and areas of specialization
- More specialization with several different cores
- An overall common umbrella under which various "specialized" disciplines will branch out as well as the addition of other emerging ones
- Branched in focus, more heterogeneous, with an increase in number of sub-specialties from what currently exists
- Perhaps more specific disciplines the way traditional engineering programs have evolved

---

# And the rest

- 179 responses on themes with fewer than 10 each, themes such as
  - automated
  - mobile
  - project-based
  - self-education
  - experiential
  - accreditation
  - creativity
  - flexible
  - etc

---

# And the best

- Less programming
  - Less programming and more strategic
  - More emphasis on information, less emphasis on programming
  - Easy to predict: lot less programming
- More programming
  - Programming will be needed by the majority of people
  - The focus within computer science programs of mastery of the software development process will increase
  - About the same, but with even less focus on basics, and more on coding

# Can't we just go back to the way it was?

- It appears that the fundamentals of computer science are not addressed well by modern curricula
- I fear that there is too much emphasis on programming and a failure to address both the theoretical underpinnings (for example, how many computing students know what a Turing machine is?) and to take an unbiased look at the situation without being unduly influenced by current fads, popular languages, etc. The field changes quite rapidly - I've been in it since the early 1960's and have seen many things come and go (anyone remember Algol? Pascal?). I fear that the fundamentals are often lost in the attempt to cover the latest concepts.