Comments on the Canadian Computing Curriculum

Troy Vasiga
Chair, Canadian Computing Competition
University of Waterloo

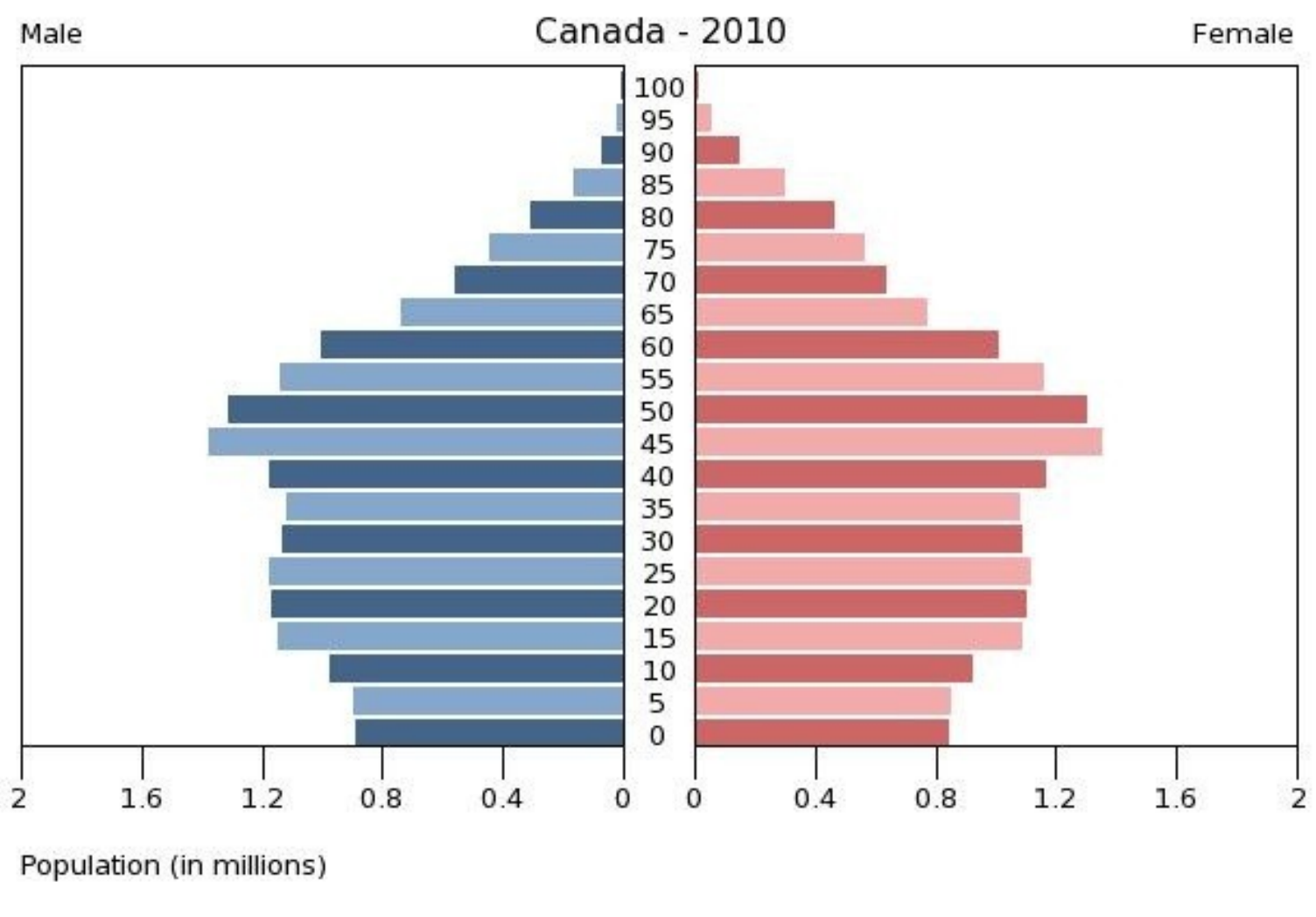
Outline

- Demographics
- Educational responsibilities
- Theory
  - Curriculum documents for province of Ontario
- Practice
  - Visiting classrooms
  - CCC Grading
- Interventions
  - Beaver
  - CS Circles
  - CCC On-Line

Demographics

- Canada in a nut shell
  - Population: 34,000,000
  - 2,400,000 students who are 10-14 years old
    - Later elementary students, early secondary school
  - 2,400,000 students who are 14-19 years old
    - Later secondary school students up to early university students
  - 76.1% of students attain a secondary school diploma
  - 53.4% of students attain a post-secondary diploma

Demographics: Population by age



Educational Structure

- Provincial domain (10 provinces, 3 territories)
  - Funding
  - Accreditation
  - Curriculum
- "Equalization" payments made to lessen the disparity between "have" and "have not" provinces
  - Still, very large differences in quality of teachers and students between provinces

Demographics

- Canadian Computing Competition
  - 2500 students write the competition each year
  - 1500 students write a "Junior" competition
    - Most difficult tasks involve recursion or complicated array manipulation
  - 900 students write the "Senior" competition
    - Most difficult tasks deal with dynamic programming (close to IOI level of difficulty)

Theory: Ontario Curriculum

- Each subject has its own curriculum document (math, science, english, etc.)
- The document for Computer Science can be found at

  `www.edu.gov.on.ca/eng/curriculum/secondary/computer.html`
- There are five CS courses:
  - ICS2O
  - ICS3U/ICS3C
  - ICS4U/ICS4C

Theory: Ontario Curriculum

- ICS2O ("O" means Open, 2 is Grade 10)
  - This course introduces students to computer programming.
  - Students will plan and write simple computer programs by applying fundamental programming concepts, and learn to create clear and maintainable internal documentation.
  - They will also learn to manage a computer by studying hardware configurations, software selection, operating system functions, networking, and safe computing practices.
  - Students will also investigate the social impact of computer technologies, and develop an understanding of environmental and ethical issues related to the use of computers.

ICS20: Understanding Computers

- A1. describe the functions of different types of hardware components, and assess the hardware needs of users;
- A2. describe the different types of software products, and assess the software needs of users;
- A3. use the basic functions of an operating system correctly;
- A4. demonstrate an understanding of home computer networking concepts;
- A5. explain the importance of software updates and system maintenance to manage the performance and increase the security of a computer.

ICS20: Introduction to Programming

- B1. describe fundamental programming concepts and constructs;
- B2. plan and write simple programs using fundamental programming concepts;
- B3. apply basic code maintenance techniques when writing programs.

ICS20: Programming Goals

- B2.3 write keyboard input and screen output statements that conform to program specifications;
- B2.4 write a program that includes a decision structure for two or more choices
- B2.5 write programs that use looping structures effectively
- B2.6 explain the difference between syntax, logic, and run-time errors;
- B2.7 compare and contrast the use of different programming environments to solve the same problem (e.g., a solution developed in a programming language versus one developed using a spreadsheet).

ICS20: Computers and Society

- C1. describe key aspects of the impact of computers and related technologies on society;
- C2. describe computer use policies that promote environmental stewardship and sustainability;
- C3. describe legal and ethical issues related to the use of computing devices;
- C4. describe postsecondary education and career prospects related to computer studies.

ICS3U/ICS3C

- Add one-dimensional arrays and subprograms
- Discuss software design and software life-cycle
- Discuss environmental aspects and career prospects of computer studies

ICS4U/ICS4C

- Type conversion and modular design
- Algorithms
- Build a project using the software life cycle
- Ethical issues and environmental stewardship

Other comments

- Ontario is very similar to British Columbia and Alberta in terms of computing curriculum
- Not a lot of integration or overlap between computing curriculum and mathematics curriculum
  - This is a bit more enhanced in B.C.
- Formerly, computing curriculum fell under the "Mathematics" department: now, it falls under the "Technology" department

Theory and Practice

- *In theory there is no difference between theory and practice. But, in practice, there is.*

  Jan L.A. van de Snepscheut

Practice: School Visits

- Part of my role in the CEMC (Centre for Education in Mathematics and Computing) is to visit secondary school math and computing classes
- Generally, the CS level is very low
- Even in ICS3U courses, students have difficulty reading input and writing a simple if statement

Practice: Teacher Conferences

- I also attend and give lectures at Teacher Workshops for secondary school teachers
- The vast majority of schools do not offer all five ICS courses
  - Most offer ICS2U
  - A few offer one ICS3, which combines both ICS3U and ICS3C (i.e., it is ICS3C)
  - Very few offer ICS4U
  - I have yet to hear of ICS4C being offered

Practice: CCC Grading

- The CCC Stage 1 contest is graded in schools by teachers
- There are a surprising number (10-15 schools) each year for which the teacher does not know how to compile or test the student programs

Bridging the Gap

- Need to get teachers to:
  1. Learn the material
  2. Encourage students to form a strong community of knowledge
  3. Get out of the way

CCC Grading

- We will use automatic grading at CCC Stage 2 this year (i.e., next week) and in Fall 2011
- Follow the IOI 2010 (immediate feedback)

CS Circles Online

- Starting in Fall 2011, we will have a self-guided, online tutorial
- Students can work through various computing concepts at their own pace, with immediate feedback
- Write small Python code programs
- Sneak Peek

Beaver contest

- Give students a reason to equate problem solving, mathematics and computer science

Conclusions

- The curriculum sets out very good, albeit high goals
- Currently, the majority of students and teachers do not find it attainable
- There are very motivated students
- Other organizations (teacher training schools, CEMC) need to provide more resources for both teachers and students