Surveying Initiatives Aimed at Increasing Female Participation in Computer Science

Shenyun Wang and Andrea Bunt

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
Contact: bunt@cs.umanitoba.ca

Date: August, 2017

Abstract / Overview

The following report describes the results of a survey that we conducted during the summer of 2017 on different strategies being employed by other Universities to address the gender imbalance in their Computer Science programs. In conducting the survey, our goal was to compare and explore the variety of strategies being used and, when possible to gain insight on which strategies have seen the most success. In the next section we describe the schools that we surveyed and our general method of obtaining information on their initiatives. We follow this description of our method with a summary of our findings. Detailed breakdowns of each institution's initiatives can be found in the appendices. We conclude with a discussion of the North American BRAID funding initiative and our final reflections.

General Survey Methodology

The following study compares and explores the variety of strategies utilized by other universities to address gender imbalances in Computer Science programs. A total of 19 prestigious universities were selected for the purposes of this survey; including the U15 Group of Canadian Research Universities and University of Manitoba, and three American institutions widely known for their continued outreach initiatives and success on this issue. Observations were recorded based on three elements: the existing gender ratio in the Computer Science departments at each institution, activities of existing Women in Computer Science (WICS) groups (when available), as well as aspects of the intuitions' curriculums that appeared to be designed around the interests and needs of female students. Other observations were obtained by attempting to find anything about women on the Computer Science department webpage of each university; this is sometimes a good indicator of an institutions' efforts of attracting more females. However, for a majority of the institutions, in order to find the WICS page and other useful articles, inputting the institution name and 'women in computer science' into the search engine was necessary. We also examined faculty member web pages to estimate the ratio of male to female instructors in each. Institutions that did not provide the male to female student ratio were contacted to acquire the data. The most useful sources used were research papers written by members of the institution which analyze the lack of women in Computer Science and what they themselves have done in regards to the issue.

Findings

Observing the difference in the ratios of females to males majoring in Computer Science among the institutes studied, combined with the various methods each institution applied, it is quite discernible that there are multiple effective ways to approach the underrepresentation of females in Computer Science. We documented available women groups (and their roles), curriculum re-design initiatives, and gender statistics for each University. For comparison purposes, these findings are summarized in tables in Appendix B and C. Some common and effective methods used by these institutions include: establishing women's groups and mentoring programs, modification of introductory courses, and inviting female speakers to talk to students. We elaborate on these strategies below.

Known Gender Statistics

Statistics Canada states that in the year 2014-2015 women accounted for only 25.5% of the enrolment in mathematics, computer and information sciences [5]. This ratio is reflected in the statistics available from the institutions included in this survey. Out of the Canadian universities that we examined, the highest ratio of females registered in Computer Sciences belongs to Queen's University at 36%, followed closely by the University of British Columbia (UBC) at 34%. McGill University reported a 27% female enrollment, the University of Toronto reported a 23% female enrolment, and the University of Waterloo a 20% female enrolment. The remaining U15 schools for which we received data were at 18-17%, whereas the University of Manitoba has been at the 10-11% mark for the last few years. Out of the three American universities chosen for the purposes of this report, Carnegie Mellon University (CMU) boasts a 40% female enrolment, while Harvey Mudd College maintains a range of 35-40%. The University of North Carolina at Charlotte, a school that has only recently invested resources into improving gender ratios, currently has a 16 ratio%. Additionally, it was noted that higher percentages of female instructors in the department appeared to positively correlate with the percentage female students in that department (See Appendix A for a complete list of gender ratio statistics). Although this study's purpose is not to prove this correlation or to imply a causation, there are other studies led by other institutions that support the idea that female students feel more comfortable seeking assistance from female superiors than their male counterparts [27].

Support Structures

Female-Centric Groups

Female-centric groups at each institution can consist of female instructors, other female students or a combination of both. These groups play a big role in creating a welcoming and supportive community for women in computer sciences and other technology-related fields. The women's group is generally responsible for recruiting and expanding the group. They do this by holding events for bonding, events that encourage women to continue in the field, as well as organizing support computer programming. A detailed description of each institution's female group can be found in Appendix B.

One of the more popular programs organized by WICS (Women in Computer Science) committees from several of the institutions studied happen to be mentorship programs that connect new students with either senior students or female Computer Scientists working in industry. This not only provides guidance to students, but also builds a larger network of women in Computer Science from various stages of academic or career-focused endeavours. At the University of McGill, the program is called the CSter program, pairing student technologists with less experienced students every semester based on experience level, skills, personality, and interests. The big CSter acts as a source of advice on any computer science related problem the little CSter may have [39]. Most of the interactions happen electronically, but there are events that give the CSter's opportunities to meet. McGill also has an Industry Mentorship program which pairs students with professional women to receive valuable advice and guidance [39]. Simon Fraser WICS provides a small-scale mentorship event once a semester to join a female faculty member with female students to bond over lunch [46]. CMU expands on their mentorship program by dividing the sister pairing into undergrads, grads, and PhD students [47]. This not only aims to help women new to Computer Science, but also encourages and supports women pursuing graduate studies, where the numbers are often even lower than in the undergrad program.

The bonding of females in Computer Science, who may have difficulty connecting through regular classroom interaction given their sparsity, is supported by WICS groups through the hosting of various socials and parties. The Waterloo committee hosts a variety of events including dinner with female professors, tea parties, a holiday party, and industrial events. Parties are a great way for female students to socialize, build friendships and to build a sense of community for females within Computer Science. On the CMU webpage, they list effective methods they have used to attract female students to join the WICS committee, one of them being the annual ice cream social held at the beginning of the school year to explain the program and activities that are offered [1]. The committee also plans various sister events like movie nights, dessert socials, and events like ice skating, and concludes that the key is to organize activities that are helpful and entertaining at the same time [1].

One of the most important programs offered by WICS are prep sessions, and volunteer opportunities. The Waterloo committee provides various volunteer opportunities like holding office hours, working on the WICS linux systems and website, running workshops and educational outreach events, or leading the big CSters program [60]. Volunteer opportunities not only help build up experience, but also reinforce a sense of belonging to the department and being capable of making a difference. This group also organizes technical interview workshops for female students three times a semester to help females to feel more confident in their interviews. Western Ontario provides an intern panel in addition to the interview prep, which is a series of questions and answers with current and past students who have found tech jobs. Similarly, CMU has a yearly event called "passing the torch" where graduating students talk about how they made it through the Computer Science program, what worked and what didn't, and advice on courses and professors [22]. This event reminds students that others have gone through what they are experiencing, and that success is attainable for themselves when confidence levels are lower, especially in the first two years of the program [22].

The university of Ottawa has a combined female group for all the sciences and engineering called Women In Sciences and Engineering (WISE). Due to the diversity of this group, the main events organized are the inspiring women series events where working female of sciences or engineering are invited to speak to students. Aside from the women series, WISE uses their website as an interactive blog to publish stories that students submit, monthly blog themes, and organize contests. In the past, there were monthly themes that females of sciences and technology could write about and submit to the blog to be posted, such as, "How I got into Science/Engineering and why I stayed", or "stress management tips". The blog once held a contest in February complementary to the month of love titled, "A love letter to my program", for women to express love for their program in science [33]. Aside from entertaining students, this also reminds students the wonderful aspects of their program despite the hardships within it. The University of Manitoba has a similar WISE group.

Funding to Attend Grace Hopper

Some of the institutions that we survey provide funding for female students to attend the annual Grace Hopper, which is an eye-opening experience that has attracted students beyond the introductory courses and into the program itself. It has in fact been shown that women who attend the Grace Hopper Conference are more likely to stay in Computer Science because of how inspiring the conference can be [24]. From the institutions studied, Harvey Mudd sends the highest number of roughly 30 students [30], CMU sends at least 6 students [31], Dalhousie university sends 2 students through the CORe CS group [25], McGill sets a goal of 2-4 if funding permits through an online donation system [44], and McMaster, UNCC, and University of Alberta each offer a scholarship for one student to attend the conference. The importance and influence of the conference is well reflected through the efforts of these institutions despite expenses involved.

Curriculum

A common characteristic of post-secondary institutions with a more desirable ratio of female students in Computer Science is the modification of the introductory curriculum. Many studies have shown that females are often more concerned about the social context and applications than males [27], hence the traditional way of teaching Computer Science (focusing on programming and syntax first) is far more appealing to men than it is to women. In response, universities like McGill, Simon Fraser, Dalhousie, Queen’s, Western, and Saskatchewan have introduced application and history courses to inform students what Computer Science is all about, beyond the stereotypes and stigmas that revolve around the field. Simon Fraser has a course titled, *Computers and Activity of People* with the description of educating computer literacy and appreciation, and how computers affect people’s lives (See Appendix C.5 for more details). Although courses like this are usually optional, they have been shown effective in drawing new students into computer science when they may have chosen the course only as an elective [49]. The Computer Science department at the University of Manitoba is introducing a new version of its first-year service course (COMP 1500), with a similar intent of focus on Computer Science applications.

Another major factor that hinders women from choosing Computer Science is that women tend to have less programming and computer experience entering university than most male peers. This lack of experience in turn leads to lower confidence levels for females surrounded by male classmates that demonstrate an advantage in both experience and knowledge [27], reinforcing the stigma that Computer Science is for males. More than half (Queen’s, UofT, Simon Fraser, U of Calgary, U of Saskatchewan, Western Ontario, Harvey Mudd, CMU) of the institutions that were studied had structured their first-year Computer Science courses with different entry points (Detailed information about each institution’s curriculum can be found in Appendix C). Simon Fraser for example, has two paths for first year students; one for beginners, and the other for more experienced students or students that intend to specialize in software development. The path for beginners starts off with a course that requires no programming background and then branches into two courses that teach the same content but at a different pace. University of Saskatchewan also redesigned their first-year Computer Science with two entry points: one for students with no background knowledge, and one for those with some background knowledge. This structure allows students with no background knowledge to learn at a less overwhelming pace with classmates who are at similar levels, while not hindering students that are looking for something more challenging. University of Calgary and CMU offer preparation courses for learning Unix, which is commonly used throughout Computer Science courses, but another tool female students tend to enter the program with no knowledge of. Such a course can help prepare students and bring them up to a level that will allow efficient learning of critical materials without the weight of struggling to learn how to use tools to help complete course material. CMU, Montreal, Western, McGill, Simon Fraser, and Dalhousie also offer intro courses for web programming or app development for the web to further help students prepare useful skills throughout university and their career. From the listed institutions, Simon Fraser University has a significantly greater variety of skills and courses available to first year students (See Appendix C.5 for detailed course list). In addition to a course on computer literacy and appreciation, there are also courses on databases and spreadsheets, communicating technical information, application of engineering and computing on society, programming in visual basics, web application development, websites, and animated programming. Having this variety not only prepares students for future courses, but also allows first year students that are still exploring different fields to have a better exposure to various sides of Computer Science, where they may find one aspect that catches their interest.

Queen's, Calgary and Western provided opportunities for individuals from other disciplines to either combine their degree with Computer Science or learn about programming that complements their work. Western Ontario offers concurrent degrees with a Computer Science degree such as the Honors Business Administration (HBA) from Ivey, Juris Doctor from Western Law, Bachelor of Music from the Don Wright Faculty of Music, Nursing Degree from Western [57], etc. This variety of options reflect the diversity Western Ontario desires to attract into the Computer Science department and their efforts to do so (Course list can be found in Appendix C.14).

Another university that has taken great measures to improve their female ratio is the University of North Carolina Charlotte. UNCC has taken a unique approach to draw in more females by creating a hands-on curriculum called FemmTech, open to both males and females (Appendix C.17). The course aims to help females deal with stigmas and sexism by inviting female tech speakers to speak about their passion for Computer Science, and how they dealt with gender-related challenges that they faced. Some programming is involved, but a big portion of the course involves following active bloggers who write about gender and tech issues, and in turn creating their own blog about the issues and what they're learning [6]. As the course has been implemented only recently, the impact of this course is yet to be known.

North American Funding Initiative: BRAID

We conclude this report with a discussion of BRAID (Building, Recruiting, And Inclusion for Diversity), a North-American funding initiative that aims to improve female participation and racial diversity in Computer Science by providing resources to implement some of the initiatives described in the previous sections.

Modelled after a set of "Beacon" schools (schools like CMU and UBC that have been successful at increasing diversity of undergraduate computing programs), BRAID is currently funding 15 participating institutions to discover the most effective structure to increase the ratio of women and minorities in Computer Science. The participating institutions receive funding of $30,000 / year for 3 years [24] to implement 3 out of 4 requirements, which include, modifying introductory courses to make them more appealing and less intimidating to underrepresented students, leading outreach programs for high school teachers and students, building confidence and community among underrepresented students, and developing or promoting joint majors in areas like Computer Science and biology that are attractive to underrepresented students [32]. The funding that BRAID provides has also allowed hundreds of females attend the Grace Hopper Conference (described above).

Conclusions

The findings of our survey indicate that a number of schools have begun to make concentrate efforts to improve their gender ratios in their Computer Science programs. It is also clear that these efforts required resources and dedicated faculty time. For example, curriculums have been re-designed, funding has been provided for events and travel, and formal Women in CS committees have been established (and recognized as part of an expected internal service load). The BRAID initiative, which provides $30,000 / year for 3 years is potentially a good reflection on both resource requirements and initial time commitment required.

References

1. About Us. (n.d.). Retrieved July 28, 2017, from https://www.women.cs.cmu.edu/Who/About/

2. Academic Calendars. (n.d.). Retrieved August 08, 2017, from http://academiccalendar.dal.ca/Catalog/ViewCatalog.aspx?pageid=viewcatalog&catalogid=69&chapterid=3521&loaduseredits=False

3. Academics. (n.d.). Retrieved August 08, 2017, from http://www.ucalgary.ca/cpsc_info/contact-us/directory/1-46929

4. All CSD Faculty. (n.d.). Retrieved August 04, 2017, from https://www.csd.cs.cmu.edu/directory/faculty

5. Canada, G. O. (2016, November 23). Canadian postsecondary enrolments and graduates, 2014/2015. Retrieved July 28, 2017, from http://www.statcan.gc.ca/daily-quotidien/161123/dq161123b-eng.htm

6. CCI initiative to tackle diversity in computing. (2016, July 20). Retrieved July 27, 2017, from http://inside.uncc.edu/news-features/2016-07-20/cci-initiative-tackle-diversity-computing

7.Computer Science. (n.d.). Courses. Retrieved August 08, 2017, from http://calendar.artsci.utoronto.ca/crs_csc.htm#ASMAJ1689

8. Computer Science. (n.d.). Faculty. Retrieved August 08, 2017, from http://web.cs.toronto.edu/people/faculty.htm

9. Computer Science CPSC. (n.d.). Retrieved August 08, 2017, from https://www.ucalgary.ca/pubs/calendar/current/computer-science.html#3713

10. Computer Science Department Courses. (n.d.). Retrieved August 08, 2017, from http://coursecatalog.web.cmu.edu/schoolofcomputerscience/courses/

11. Computing Science. (n.d.). Courses. Retrieved August 08, 2017, from https://www.ualberta.ca/computing-science/undergraduate-studies/course-directory

12. Computing Science. (n.d.). Retrieved August 04, 2017, from https://www.ualberta.ca/computing-science/faculty-and-staff/faculty

13. Computing Science Courses. (n.d.). Retrieved August 08, 2017, from https://www.sfu.ca/students/calendar/2017/fall/courses/cmpt.html

14. Condon, K. (2016, October 28). I am woman, hear me code - Faculty of Science. Retrieved July 28, 2017, from https://www.ualberta.ca/science/about-us/contours/2016-fall-contours/2016/october/i-am-woman-hear-me-code

15. Course Descriptions. (n.d.). Retrieved August 08, 2017, from https://catalog.uncc.edu/content.php?filter%5B27%5D=-1&filter%5B29%5D=&filter%5Bcourse_type%5D=1946&filter%5Bkeyword%5D=&filter%5B32%5D=1&filter%5Bcpage%5D=1&cur_cat_oid=20&expand=&navoid=1348&search_database=Filter#acalog_template_course_filter

16. Course Listings. (n.d.). Retrieved August 08, 2017, from http://academiccalendars.romcmaster.ca/content.php?filter%5B27%5D=COMP%2BSCI&filter%5B29%5D=&filter%5Bcourse_type%5D=-1&filter%5Bkeyword%5D=&filter%5B32%5D=1&filter%5Bcpage%5D=1&cur_cat_oid=7&expand=&navoid=558&search_database=Filter&filter%5Bexact_match%5D=1#acalog_template_course_filter

17. Courses@CS. (n.d.). Retrieved August 08, 2017, from https://www.cs.mcgill.ca/academic/courses

18. Faculty. (n.d.). Retrieved August 08, 2017, from http://www.cs.mcgill.ca/people/faculty

19. Faculty Directory. (n.d.). Retrieved August 08, 2017, from http://cs.uncc.edu/people/cs-faculty

20. Faculty & Staff. (n.d.). Retrieved August 08, 2017, from https://www.dal.ca/faculty/computerscience/faculty-staff.html

21. Favron, C. (2017, March 10). Cecile Favron. Retrieved July 28, 2017, from http://www.the-peak.ca/2017/03/gender-disparities-in-engineering-and-computing-science-at-sfu-and-its-effects-on-female-students/

22. Frieze, C., & Blum, L. (2002, June). Building an Effective Computer Science* Student Organization: The Carnegie Mellon Women@SCS Action Plan. Retrieved July 28, 2017, from https://www.cs. .edu/~cfrieze/women@scs/paper.html

23. Full-Time Faculty. (n.d.). Retrieved July 28, 2017, from https://www.csd.uwo.ca/people/faculty/full_time_faculty.html

24. Galatzer-Levy, J. (2014, September 25). UIC helps bring women, minorities into computer science. Retrieved July 28, 2017, from https://news.uic.edu/uic-to-help-bring-women-minorities-into-computer-science

25. Grace Hopper Conference (GHC) 2016 [PDF]. (2016). Dalhousie University.

26. Haikara, N. (2017, June 7). Women's research inquiry [E-mail to the author].

University of Toronto: Toronto campus

27. Hancock, M. S., R. D., & McGrenere, J. (2004). Focus on Women in Computer Science. Retrieved July 04, 2017, from https://www.cs.ubc.ca/wccce/program04/Papers/mark.html

28. HARVEY MUDD COLLEGE. (n.d.). Course-description. Retrieved August 08, 2017, from https://www.cs.hmc.edu/program/course-descriptions/

29. HARVEY MUDD COLLEGE. (n.d.). Retrieved August 04, 2017, from https://www.cs.hmc.edu/people/

30. Harvey Mudd College Women. (2014, January 16). Retrieved August 04, 2017, from https://www.hmc.edu/about-hmc/2013/09/24/harvey-mudd-college-women-think-big-at-grace-hopper-conference/

31. INI: The Information Networking Institute. (n.d.). Retrieved July 28, 2017, from http://www.ini.cmu.edu/news/2015/11/gracehopper.html

32. Kasda, R. (2016, September 14). Building, Recruiting And Inclusion for Diversity ... Retrieved July 28, 2017, from https://www.bing.com/cr?IG=BE9880DFDDDD4630ABD76F1F613D6A55&CID=2EEBD61AE42167360C96DCD7E527661F&rd=1&h=Wo7hBW3x1AAS0lM7nRVcxJkIb8Wf_dytZbwf8_yp1hQ&v=1&r=https%3a%2f%2fcra.org%2fbuilding-recruiting-inclusion-diversity-initiative-seeking-affiliate-schools%2f&p=DevEx,5064.1

33. M. (2011, February 14). Announcing FEBRUARY CONTEST: “A Love Letter To My Program”. Retrieved July 28, 2017, from https://uottawawise.wordpress.com/2011/02/14/announcing-february-contest-a-love-letter-to-my-program/

34. McWiCS. (n.d.). Retrieved July 28, 2017, from https://www.mcwics.com/

35. More women are studying computer science at Waterloo. (2017, June 16). Retrieved July 28, 2017, from https://uwaterloo.ca/stories/more-women-are-studying-computer-science-waterloo

36. Olafson, K. (2015, September 08). The Girl Code: How Calgary Is Trying to Foster Diversity In STEM. Retrieved July 28, 2017, from http://www.avenuecalgary.com/City-Life/The-Girl-Code-Calgary-STEM-Technology-Diversity/

37. People in CAS. (n.d.). Retrieved August 04, 2017, from http://www.cas.mcmaster.ca/cas/0template1.php?501

38. Percentage of Women in First-year Class at Carnegie Mellon University [Advertisement]. (2016, September 2). Retrieved July 3, 2017, from https://www.cmu.edu/news/stories/archives/2016/september/images/cmu_factsheet.pdf

39. Programs and Events. (n.d.). Retrieved July 28, 2017, from https://www.mcwics.com/programs/

40. Queen's WISC . (n.d.). Retrieved July 28, 2017, from http://qscwisc.weebly.com/

41. Rawcliffe, R. (2017, April 28). Dal 200 planning profile: Faculty of Computer Science. Retrieved July 28, 2017, from https://www.dal.ca/news/2017/04/28/dal-200-planning-profile--faculty-of-computer-science.html

42. School of Computing at Queen's University. (n.d.). Faculty. Retrieved July 28, 2017, from http://www.cs.queensu.ca/people/faculty.php

43. School of Computing at Queen's University. (n.d.). Undergraduate Courses. Retrieved August 08, 2017, from http://www.cs.queensu.ca/students/undergraduate/courses/

44. Seeds of Change | McGill University. (n.d.). Retrieved July 28, 2017, from https://www.mcgill.ca/seedsofchange/project/grace-hopper-initiative

45. Simon Fraser University: Engaging the World. (n.d.). Faculty. Retrieved August 08, 2017, from http://www.sfu.ca/computing/people/faculty.html

46. Simon Fraser University: Engaging the World. (n.d.). Mentor Lunch. Retrieved July 28, 2017, from http://www.sfu.ca/computing/wics/events/mentor-lunch.html

47. Sisters Mentoring Programs. (n.d.). Retrieved July 28, 2017, from https://www.women.cs.cmu.edu/What/BigLittleSisters/

48. Social Programs. (n.d.). Retrieved July 28, 2017, from https://www.women.cs.cmu.edu/What/Social/

49. Staley, O. (2016, August 22). Harvey Mudd took on inequality and now has more women than men in computer science. Retrieved July 28, 2017, from https://qz.com/730290/harvey-mudd-college-took-on-gender-bias-and-now-more-than-half-its-computer-science-majors-are-women/

50. The University of British Columbia. (n.d.). Courses. Retrieved August 08, 2017, from https://courses.students.ubc.ca/cs/main?dept=CPSC&pname=subjarea&req=1&tname=subjareas

51. The University of British Columbia. (n.d.). Faculty. Retrieved August 08, 2017, from https://www.cs.ubc.ca/our-department/people/faculty

52. The University of British Columbia. (n.d.). Women Statistics.

53. The University of Saskatchewan. (n.d.). Computer Science. Retrieved August 08, 2017, from https://www.cs.usask.ca/students/courses/

54. The University of Western Ontario : Computer Science (S). (n.d.). Retrieved August 08, 2017, from http://www.westerncalendar.uwo.ca/2016/pg900.html#34169

55. UOttawa. (n.d.). Computer Science (CSI). Retrieved August 08, 2017, from http://catalogue.uottawa.ca/en/courses/csi/

56. Undergraduate Computer Science courses. (n.d.). Retrieved August 08, 2017, from https://cs.uwaterloo.ca/current/courses/course_descriptions/

57. Undergraduate Students. (n.d.). Retrieved July 27, 2017, from https://www.csd.uwo.ca/prospective_students/undergraduate_students/index.html

58. University Of Manitoba. (n.d.). Retrieved August 04, 2017, from https://www.cs.umanitoba.ca/people/

59. University of Manitoba: Catalog Term. (n.d.). Retrieved August 10, 2017, from https://aurora.umanitoba.ca/banprod/bwckctlg.p_disp_dyn_ctlg

60. Women in Computer Science. (n.d.). Retrieved July 28, 2017, from http://wics.uwaterloo.ca/volunteer/

61. Women in Technology Society. (n.d.). Retrieved July 28, 2017, from http://wits.csd.uwo.ca/

62. Yates, C. (2014). Women faculty, now and in the future: Building excellence at McMaster University (Rep.). http://www.mcmaster.ca/vpacademic/documents/Yates_Report_on_Gender_Equity_January_2014.pdf

63. Ye, Y. (2017, June 18). Women's research Inquiry [E-mail to the author].

University of Toronto: Mississauga

Appendix

Appendix A: Gender Statistics

Universities	Female Prof ratio	Female Student ratio
Dalhousie University	17% [20]	17% [36]
McMaster's University	11.5% [37]	33% [62]
Queen's University	24.5% [42]	36% [40]
Simon Fraser University	12% [45]	18% [21]
University of Alberta	17% [12]	20% [14]
University of British Columbia	20% [51]	28.7% [52]
University of Calgary	14% [33]	17% [38]
University of McGill	17% [18]	27% [41]
University of Manitoba	12.9% [58]	11%
University of Montreal	Unknown	unknown
University of Ottawa	Unknown	Unknown
University of Saskatchewan	Unknown	Unknown
University of Toronto	21% [8]	17% Mississauga [63] 22% Toronto [26]
University of Waterloo	13% [23]	20% [35]
Western University	17.6% [32]	18% [61]
Carnegie Mellon University	10.9% [4]	48.5% [38]
Harvey Mudd College	52.9% [29]	55% [49]
University of North Carolina Charlotte	38.5% [19]	16% [6]

Collected from 2014-2017

Appendix B: Female Groups

B.1 Summary Table of Female Group Involvement

University	WICS group	Outreach	Mentorship	Events (socials, parties, etc)	Workshop	Volunteer opportunities	Successful Females involved
Dalhousie University	WIT			✓			
McMaster University							
Queen’s	✓			✓			
Simon Fraser University	✓	✓	✓	✓	✓		✓
University of Alberta							
University of British Columbia	✓	✓	✓				✓
University of Calgary							
University of Manitoba	WISE	✓		✓	✓	✓	✓
University of McGill	✓		✓				✓
University of Montreal							
University of Ottawa	WISE			✓			✓
University of Saskatchewan							
University of Toronto				✓			✓
University of Waterloo	✓	✓	✓	✓	✓	✓	✓
Western University	WIT			✓	✓		✓
Carnegie Mellon University	✓		✓	✓			✓
Harvey Mudd							
University of North Carolina Charlotte			✓				

B.2 Dalhousie University

Women in technology society is a group targeted at women studying a STEM program with a focus on technology.

Events

Various get together events are hosted such as Food Truck Party's, pizza delight night, and socials.

B.3 Queen's University

Women in the school of computing represents all women at Queen's university including students, faculty and staff in the computer science department.

Events

End of the semester celebrations consisting of movies and ice cream is organized.

B.4 Simon Fraser University

WiCS at Simon Fraser welcomes students of all genders to join the committee.

Mentorship

The Mentor Lunch gathers a female faculty member and a few female students for discussion of interests over lunch.

Outreach

Try/CATCH (Computing and Technology Conference for Her) is a one day event for high school girls from grades 8-12. Through interactive and hands on workshops, the WiCS team hopes to help these school girls to develop an interest in computers and technology.

Events

In collaboration with WiE (Women in Engineering), WiCS organizes both, annual hackathons where students gather to develop an app or a site that proposes a solution to a theme, and Networking Nights to meet successful women in technical positions. End of semester Beach BBQ and fireworks.

Workshops

Database and scholarship workshops are offered by the women's group.

B.5 University of British Columbia (UBC)

The female group at UBC is called Focus on Women in Computing Society, and their goal is to increase the participation of women and gender minorities in computer science.

Mentoring

The Tri-mentoring program matches junior undergrads/grads with senior undergrads/grads, and the senior mentees are in turn matched with industry or faculty mentors. This program is inclusive to all genders with over 50% female participants.

Outreach

GIRLsmarts4Tech is an annual one day workshop for grade 6 and 7 girls, inspiring them to explore technology through a variety of programming activities.

B.6 University of Manitoba

WISE Kid-Netic energy is an organization that offers various science, engineering and technology interactions and exposures to girls in kindergarten through grade 12.

Workshops

Workshops offered includes a variety of robots, tours, mentor events, community outreach projects, basic software coding, and more. Girls may sign up for a certain workshop which will run for a varying number of weeks.

Volunteer

WISE offers females a chance to lead younger girls and help them experience the excitement of engineering and sciences.

Events

Engineer events are organized for middle to high school girls to explore the field.

B.7 University of McGill

The computer science female committee at McGill is called McWiCS (McGill Women in Computer Science.

Mentoring

The CSter program matches seniors and PhD students with junior students based on interests, personality, and experience. Casual events are hosted throughout the year for the CSter pair to encourage bonding.

Industry mentorship pairs students with professional women who volunteer themselves as mentors for the Industry Mentorship program. Students are expected to meet up with their industry mentor at least once a month for valuable discussion.

Events

The committee prepares workshops that help students explore different possibilities in computer science. They also host events like study sessions, ice cream socials, casual hack nights, freshmen socials, and dinners.

B.8 University of Ottawa

WISE of Ottawa is a group for all females in sciences and engineering. The executive team consists of females in various programs like computer science, biology, engineering, etc.

Events

Blog contests are held often to encourage students to reflect on the positives of their program. Coffee Socials and Lunchtime workshops with professors or invited guest speakers are also available for females to attend.

B.9 University of Toronto

The WICS group at U of Toronto is an informal group of female students who gather for monthly lunches to socialize and bond. Sometimes guest speakers are invited for discussion.

B.10 University of Waterloo

The Waterloo WICS branches into two committees one for undergraduates and one inclusive to all women in computer science.

Volunteer

The main committee offers volunteer positions for running workshops and educational outreach events, office hours for the WICS office, coding on the linux systems and website, and leading the Csters mentoring program.

Events

Events offered by the committee includes T-shirt contests, dinner with female professors, mentorship and cake night, special events with companies, tea party, holiday party's, and technical interview workshops.

Mentorship

Csters mentorship program connects women (trans and nonbinary included) through events that promote mentorship and community building.

B.11 Western University

WiT (Women in technology) supports the community of female students in Computer Science and Software Engineering.

Support

WiT offers interview prep sessions and an intern panel for students to inquire about internships from experienced students who have found jobs in the tech field. The Industry Panel session invites industry professionals to answer questions and share insights and experiences.

Events

Events WiT organizes includes small hackathons, a Christmas social, and Bi-weekly hangouts give females a chance to bond and relax.

B.12 Carnegie Mellon University

The Women@scs (school of computer science) group at CMU is an advisory council consisting of both undergraduate and graduate students.

Mentorship

There are 3 sister mentoring programs at CMU; the undergraduate Big Sister/Little Sister Program, Grad/Undergrad sisters, and PhD sisters [36].

Events

Events held by Women@scs includes class mixers which encourages women in the same graduating class to get to know each other, ice cream socials, sisters socials, graduate tea hours for graduate student bonding, and passing the torch from graduating females to new students [35].

B.13 University of North Carolina Charlotte

Mentorship

The mentoring program at UNCC helps female students advance in academic prep towards joining the technical professions and advance to positions of leadership.

Appendix C: Curriculum

C.1 Summary Table of First year courses

University	Beginner course (no experience)	Application-centric courses	Remedial courses	Interest courses	Other discipline cs courses
Dalhousie University	CSCI 1200	CSCI 1108, CSCI 1616	--	CSCI 1106, CSCI 1101	CSCI 1200
McMaster University		COMP SCI 1JC3	--	COMP SCI 1BA3	--
Queen's University	CISC 101, CISC 110	CISC P81	--	CISC 110	--
Simon Fraser University	CMPT 102, CMP 120	CMPT XX1, CMPT 105W, CMPT 106	CMPT 100, CMPT 110	CMPT XX1, CMPT 118, CMPT 150, CMPT 165, CMPT 166, CMPT 170	CMPT 118, CMPT 128, CMPT 129
University of Alberta	CMPUT101, CMPUT174	--	--	--	--
University of British Columbia	CPSC 100, CPSC 103	--	--	CPSC 103	CPSC 103
University of Calgary	--	--	CPSC 101	--	CPSC 203, CPSC 217
University of McGill	COMP 102,	COMP 189	--	COMP 189	--
University of Ottawa	--	--	--	CSI 1306	CSI 1306
University of Saskatchewan	CMPT 100, CMPT 140	CMPT 100	--	CMPT 120	CMPT 113, CMPT 116, CMPT 120
University of Toronto	CSC104H1, CSC108H1	CSC104H1	--	CSC104H1	CSC120H1, CSC121H1,
University of Waterloo	CS 105, CS 135	CS 106	--	CS 105	--
Western Ontario University	1046A/B	1101A/B	--	1033A/B	1101A/B, 1032A/B, 1037
CMU	15-104, 15-110, 15-112	15-110, 15-128	15-131, 15-151	15-128, 15-199	--
Harvey Mudd				CS 5GR	CS 5GR
UNCC	ITCS 1101	ITCS 1203, LBST 2213	ITCS 1610	ITCS 1301	--

C.2 Dalhousie University *[2]*

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CSCI 1100	Computer Science I	Math	CSCI 1202	This course provides a general introduction to computer science and the hardware and software of computers. The main focus is on programming skills and how to apply these skills in solving a variety of problems. Algorithmic concepts are stressed.
CSCI 1101	Computer Sciences II	CSCI 1100	N/A	This course focuses on programming and linear data structures.
CSCI 1106	Animated computing	N/A	N/A	This course is a hands-on introduction to two areas of computing: robotics and game design. Students will learn how to program simple robots and will use empirical methods to investigate various aspects of robotics, such as sensors and actuators, uncertainty, knowledge representation, and failure detection and recovery. Students will also learn how to design and develop simple games and will investigate aspects of game design, such as event driven frameworks and collision detection. Students will use empirical methods to investigate the player-centric aspects of game design. Students work in groups to complete small robotics and game design projects.
CSCI 1101	Social computing	N/A	N/A	A hands-on course on technologies and the underlying infrastructure for social computing, including digital collaboration media, social networks and visualization, and their social impact. Students will use various applications, such as Twitter, YouTube, and wikis to examine their functionalities, and explore infrastructure technologies including databases, computer networks and Web-servers that facilitate their execution.
CSCI 1108	Intro to experimental robotics	N/A	N/A	This course is an introduction to computing in the real world, experimental evaluation, and basic team work and project management. These themes are explored through the study of the challenges and techniques for modeling, designing, and programming robotic systems to perform complex tasks in a changing and uncertain environment.

CSCI 1110	Computer science I and II	N/A	CSCI 1100, CSCI 1101	This course provides a general introduction to computer science and the hardware and software of computers. The main focus is on programming skills and how to apply these skills in solving a variety of problems. Algorithmic concepts, linear data structures are emphasized.
CSCI 1200	Intro to computing for non-majors	N/A	N/A	This is a course of technical computer literacy. Students will be given an opportunity to become familiar with typical applications of software such as word processors, spreadsheets and database applications. Other topics will include the use of the internet, creation of web pages, and simple programming concepts. **No previous computer experience is required.** This course is open to Arts and Social Sciences and Health Education students only.
CSCI 1615	Concepts in computing	N/A	N/A	This course introduces some key concepts in computing and places them in context with a survey of applications. The skills developed in this course include research, redaction, problem solving and abstraction. The themes covered are the following: Data and storage, operating systems, networking and the Internet, database systems, artificial intelligence and computer graphics. This course is eligible to partially cover the writing requirement for students in the Faculty of Computer Science.
CSCI 1616	Applications in computing	N/A	N/A	This course covers professional aspects of the computing industry. The main themes are: project management, software engineering and computer languages. It explores how technological advances impact the workplace for non-technical people. It also includes a module on proofreading and editing. This course is eligible to partially cover the writing requirement for students in the Faculty of Computer Science.

C.3 McMaster University *[16]*

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
COMP SCI 1BA3	Intro to business information and communication technologies	Math	COMP SCI 1TA3	Business information, communication technologies, decision making, information technology tools, information literacy, introduction to databases.
COMP SCI 1JC3	Intro to computational thinking	Math	N/A	Inquiry into ideas and methods of CS, the science underlying our computational universe. Topics include what computers can and cannot do, the Internet and search engines, artificial intelligence, computer-controlled devices, and sustainability in computing.
COMP SCI 1MD3	Intro to programming	Math	Engineer course	Introduction to fundamental programming concepts: values and types, expressions and evaluation, control flow constructs and exceptions, recursion, input/output and file processing.
COMP SCI 1TA3	Elementary computing and computer use	N/A	COMP SCI 1BA, 1MD3	Organization of microcomputers and overview of computer communications; introduction to information exchange using word processing/ presentation software, the Internet and Web pages; problem solving using electronic spreadsheets and database applications.
COMP SCI 1XA3	Computer science practice and experience	Taken with COMP SCI 1MD3	N/A	Study through implementation of basic CS concepts such as data representation, recursion, computer architecture, concurrency. Hands-on application of CS concepts to formulating, analyzing, and solving problems.

C.4 Queen's University [43]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CISC P81	Computers: Applications and Implications	N/A	CISC 124	The goal of this course is to foster appreciation of the applicability of computer-based technology in diverse areas, but also to increase awareness of the limits and risks of such technology, and to increase understanding and appreciation of Computing Science as a discipline.
CISC 101	Elements of Computing Science	N/A	CISC 110, 121	Introduction to algorithms: their definition, design, coding, and execution on computers. Intended for students who have no programming experience.
CISC 102	Discrete math for computing I	N/A	CISC 203	Introduction to mathematical discourse and proof methods. Sets, functions, sequences, and relations. Properties of the integers. Induction. Equivalence relations. Linear and partial orderings.
CISC 110	Elementary computer animation	N/A	May not be taken after CISC 121	Introduction to fundamental programming concepts in the context of visual, interactive media. Students may develop applications in any domain (e.g., fine art, education, commerce, physical or social sciences) while learning about algorithms, program design, logic, program control flow, functions, testing etc. No computing background required.
CISC 121	Intro to CS I	CISC 102	N/A	Introduction to design and analysis of algorithms. Control structures: recursion, backtracking, exits. Data structures: structures, sequences, linked lists and references, binary search trees. Elementary searching and sorting. Introduction to assertions and loop invariants. Introduction to order-of-magnitude complexity. Introduction to numerical computation. Documentation, testing and debugging.
CISC 124	Intro to CS II	CISC 121	N/A	Introduction to object-oriented design, architecture, and programming. Use of packages, class libraries, and interfaces. Encapsulation and representational abstraction. Inheritance. Polymorphic programming. Exception handling. Iterators. Introduction to a class design notation. Applications in various areas.

C.5 Simon Fraser [13]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CMPT XX1	Computers and the activity of people	N/A	N/A	Concerned with computer literacy and appreciation. Applications of computing in the arts, commerce, industry, science and everyday activity. Not for students who have enrolled or obtained any other computing science course.
CMPT 100	Software packages and programming	Math	CMPT 101, 102, 103, 104, 120, 126, 128, 130	Intro to fundamentals of computer operation and programming. Use of software packages is emphasized, focusing on spreadsheets, databases, and presentation graphics.
CMPT 102	Intro to scientific computer programing	N/A	CMPT 120, 128, 130, 166	Programming course and an intro to computing concepts, structure programming, and modular design. The student will also gain knowledge in the use of programming environments including the use of numerical algorithm packages.
CMPT 105W	Process, form, and convention in professional genres	Corequisite: CMPT 106	N/A	Fundamentals of informative and persuasive communication for professional engineers and computer scientists in order to assist students thinking critically about various contemporary technical, social, and ethical issues.
CMPT 106	Applied science, technology and society	CMPT 105W	N/A	Reviews different modes of thought characteristics of science, engineering, and computing. Examines the histories and current research issues. Considers ethical and social responsibilities of engineering and computing work.
CMPT 110	Programming in visual basic	Math	Not for students with credit or enrolled in CS course at 200 level or higher	Topics will include user interfaces, objects, event-driven programming, program design, and file and data management.
CMPT 118	Special topics in computer and information tech	N/A	Not for students with credit or enrolled in CS course at 200 level or higher	Special topics of current interest to non-computing students.

CMP 120	Intro to CS and programming I	Math	CMPT 102, 128, 130, 166, 125, 129, 130, 135	Intro to computing science and programming. Fundamental concepts and terminology of computing science. Topics include pseudocode, data types and control structures, fundamental algorithms, computability and complexity, computer architecture, and history of computing science. For students with little or no programming background
CMPT 125	Intro to CS and programming II	CMPT 120	CMP 126, 129, 135	Topics include fundamental algorithms, elements of empirical and theoretical algorithmic, abstract data types and elementary data structures, basic OOP and software design, computation and computability, specification and program correctness, and history of computing science. For students with some programming background.
CMPT 126	Intro to CS and programming	CMPT 120	CMPT 125, 128, 130, 15	Topics include fundamental algorithms and problem solving, abstract data types and elementary data structures, basic OOP and software design, elements of empirical and theoretical algorithmic, computation and computability, specification and program correctness, and history of computing science. For students with substantial programming background
CMPT 127	Computing laboratory	CMPT 120 or CMPT 128, or CMPT 130	N/A	Hands-on intro to programming in C and C++. Taken with CMPT 125
CMPT 128	Intro to computing science and programming for engineers	Math	CMPT 102, 120, 130, 125, 129, 135	For students majoring in engineering or related program. Tis course introduces basic computing science concepts, and fundamentals of object oriented programming.
CMPT 129	Intro to computing science and programming for mathematics and statistics	CMPT 102 or CMPT 120	CMPT 125, CMPT 135	A second course in computing science and programming for students studying math or stats or actuarial science. Suitable for students with some programming background.
CMPT 130	Intro to programming I	Math	CMPT 102, 120, 128, 166	Intro to computing science and programming using C or C++. Topics include elementary data types, control structures, functions, arrays and strings, fundamental

				algorithms, computer organization and memory management.
CMPT 135	Intro to programming II	CMPT 130	CMPT 125, 126, 129	Topics include, techniques for designing and testing programs, use and implementation of elementary data structures and algorithms, introduction to embedded systems programming.
CMPT 150	Intro to computer design	CMPT 120	CMPT 290	Digital design concepts are presented. Topics include basic Von Neumann compute architecture, intro to assembly language programming, combinational logic design, sequential logic design.
CMPT 165	Intro to internet and world wide web	N/A	CMPT 125, 135, 170	Examine the structure of the internet and the world wide web as well as design and create web sites.
CMPT 166	And animated intro to programming	N/A	CMPT 125, 129, 135	An informal intro to programming using examples drawn f rom animation and graphics. Class design, event-driven programming or other advanced programming techniques may be introduced. No prior programming experience assumed.
CMPT 170	Intro to web app development	One of: CMPT 120, 126, 128, or 135	CMPT 118, CMPT 165	An introduction to the creation of webpages, as well as interactive websites. Creation of web-based application using a modern web application framework.

C.6 University of Alberta [11]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CMPUT101	Intro to computing	N/A	Cannot take for credit if already taken CMPUT 114, 174	Intro to fundamental concepts in computing science, including state, abstraction, composition, and representation. Intro to algorithms, logic circuits, machine architecture and other topics in elementary computing science.
CMPUT174	Intro to foundations of computing I	Math	N/A	Basic notions of state, control flow, data structures, recursion, modularization, and testing are introduced through solving problems in a variety of domains such as text analysis, map navigation, game search, simulation, and cryptography. No prior programming experience is needed.
CMPUT175	Intro to foundations of computing II	CMPUT 174	N/A	Objects, functional programming, time and memory consumption, and user interface building are explored.

C.7 University of British Columbia [50]

Course Code	Course Name	Anti-requisites	Course Description
CPSC 100	Computational thinking	CPSC 110	For students with no prior programming experience. The meaning and impact of computational thinking. Solving problems using testing and debugging. How computers work.
CPSC 103	Intro to systematic software design	CPSC 110	No programming experience expected. Computation as a tool for systematic problem solving in non-cs disciplines. Introductory programming skills.
CPSC 110	Computation, programs, and programming	N/A	Fundamental program and computation structures. Introductory programming skills. Computation as a tool for information processing, simulation and modeling, and interacting with the world.
CPSC 121	Models of Computation	math	Physical and mathematical structures of computation. Boolean algebra and combinations logic circuits; proof techniques; functions and sequential circuits; sets and relations; finite state machines; sequential instruction execution.

C.8 University of Manitoba [59]

Course Code	Course Name	Pre-requisites	Anti-requisites	Description
COMP1010	Intro to CS I	Math	COMP1012	An introduction to computer programming using a procedural high level language.
COMP1020	Intro to CS II	COMP1010	COMP1021	More features of a procedural language, elements of programming.
COMP1500	Computing: Ideas and innovation	N/A	Second year CS courses	An introduction to the topics of Computer Science and problem solving. Students will learn concepts in computer programming. Intended for students not in computer science.
COMP1013	Programmation informatique pour scientifiques et ingénieurs	Math	COMP1010, COMP1011	An introduction to problem-solving computer programming in science and engineering. Students develop algorithms for digital processing, statistical analysis and matrix operations.
COMP1021	Introduction à l'informatique II	One of COMP1011, COMP1010, COMP1013, or COMP1012	COMP1020	Intro to object-oriented programming, computer data structures and algorithms.
COMP1600	Navigating your Digital World	N/A	COMP1270	Topics related to digital society such as security, encryption and data storage, issues of social and ethical importance, and current events. Not for CS students.

C.9 University of Calgary [9]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CPSC 101	Intro to Unix	N/A	N/A	Intro to Unix operating system, including shell usage and some advanced Unix commands. Not included in GPA
CPSC 102	Advanced Unix	CPSC 219 or 233 or 235	N/A	Unix signals, processes, and file system; interprocess communication; advanced shell programming, program profiling.
CPSC 105	Intro to the analysis of algorithms	CPSC 313 and 319	N/A	Techniques to prove the correctness and measure the efficiency of algorithms
CPSC 203	Intro to problem solving using application software	N/A	Not for CS majors	Introduction to computer fundamentals; contemporary topics, such as security and privacy, and the Internet and World Wide Web. Problem solving, analysis and design using application software, including spreadsheets and databases.
CPSC 217	Intro to CS for multidisciplinary studies I	N/A	CPSC 231, CPSC 235	Introduction to problem solving, analysis and design of small-scale computational systems and implementation using a procedural programming language. For students wishing to combine studies in computer science with studies in other disciplines.
CPSC 219	Intro to CS for multidisciplinary studies II	CPSC 217	CPSC 233, CPSC 235	Emphasis on object oriented analysis and design of small-scale computational systems and implementation using an object oriented language. Issues of design, modularization and programming style will be emphasized.
CPSC 231	Intro to CS for CS majors I	N/A	CPSC 217, CPSC 235	Introduction to problem solving, the analysis and design of small-scale computational systems, and implementation using a procedural programming language. For computer science majors.
CPSC 233	Intro to CS for CS majors II	CPSC 231	CPSC 219, 235	Emphasis on object-oriented analysis and design of small-scale computational systems and implementation using an object oriented language. Issues of design, modularization, and programming style will be emphasized.
CPSC 235	Advanced intro to CS	Consent of department	CPSC 217, 219, 231, 233	An accelerated introduction to problem solving, the analysis and design of small-scale computational systems and implementation using both procedural and

				object oriented programming languages. Issues of design, modularization, and programming style will be emphasized.

C.10 University of McGill [17]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
COMP 102	Computers and computing	High school math	N/A	**A course for students with no previous knowledge of computer science**. The impact of computers on society. Web design and dynamic content. The inner workings of computers. Networking principles. Algorithm design and programming. A look at how computers store data.
COMP 189	Computers and society	N/A	N/A	How computer technologies shape social notions such as ownership, safety, and privacy. Emphasis is on computer science powering both day-today technologies and those in the news. Discussions will investigate technology and social issues in order to understand both.
COMP 199	Excursions in Computer Science	High school math	N/A	Seminar format course intended for freshmen. Possible topics are computability, complexity, geometry, vision, AI, pattern recognition, machine models, cryptography, and security and social implications of computing.
COMP 202	Foundations of programming	Math	COMP 208	Intro to programming in high level language. Intro to algorithms, data structures, modular software design, libraries, file input/output, debugging, exception handling.
COMP 206	Introduction to software systems	COMP 202	N/A	Comprehensive overview of programming in C, use of system calls and libraries, debugging and testing of code; use of developmental tools, version control systems.
COMP 208	Computers in engineering	Calculus	COMP 202	Intro to computer systems. Concepts and structures for high level programming. Numerical algorithms such as root finding, numerical integration and differential equations. Non-numerical algorithms for sorting and searching.
COMP 230	Logic and computability	Math	N/A	Propositional logic, predicate calculus, proof systems, computability Turing machines, Church-Turing thesis, unsolvable problems, completeness, incompleteness, Tarski

				semantics, uses and misuses of Godel's theorem.
COMP 250	Intro to CS	Familiarity with a high-level language	N/A	Mathematical tools, data structures, recursive and non-recursive algorithms. Abstract data types, inheritance.
COMP 251	Algorithms and data structures	N/A	N/A	Intro to algorithm design and analysis. Graph algorithms, greedy algorithms, data structures, dynamic programming, maximum flows.
COMP 273	Intro to computer systems	N/A	N/A	Number representations, combinational and sequential digital circuits, MIPS instructions and architecture data path and control, caches, virtual memory, interrupts and exceptions, pipelining.

C.11 University of Ottawa [55]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CSI 1306	Computing concepts for business	N/A	CSI 1308, CSI 1390	Introduction to computer-based problem solving from the perspective of the business world. Design of algorithms for solving business problems. Basics of computer programming in a modern programming language. Solving business problems using application packages including spreadsheets and databases. Basics of web design. Collaborative tools. Using open source software.
CSI 1308	Intro to computing concepts	Math	CSI 1306, CSI 1390	Introduction to computer based problem solving for scientific applications. Design of algorithms and algorithms descriptions. 4th generation languages. Software packages. Structured program development. Modular and object-oriented programming. Program testing.
CSI 1390	Intro to computers	N/A	CSI 1306, CSI 1308	Computing and computers. Problem solving and algorithm development. Introduction to programming. Use of application, communication, and database software.
ITI 1120	Intro to computing I	N/A	N/A	Problem solving and algorithm design. Basic principles of software engineering: structure decomposition, documentation, testing and debugging. Variable types, expressions and assignment. Conditional and iterative control structures. Modules and parameter passing. Recursion. Fundamental data structures:

				arrays, strings, matrices, records. Introduction to objects.
ITI 1121	Intro to computing II	ITI 1120	N/A	Object-oriented programming. Abstraction principles: information hiding and encapsulation. Linked lists, stacks, queues, binary search trees. Iterative and recursive processing of data structures. Virtual machines.

C.12 University of Saskatchewan [53]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CMPT 100	Intro to computing	Math	N/A	A survey of major computer science areas, combining a breadth of topics with depth via specific examples within each topic. Topics include: history of computing, computer applications, analysis and design, high level programming, computer software, computer hardware, artificial intelligence, and the social impact of computers.
CMPT 113	Computing using excel and VBA	Math	N/A	An emphasis is placed on learning many important concepts used to create useful computer programs in Excel. Examples of some concepts include arrays, procedures, and functions. **This course is mostly intended for engineering students**. Some graphical user interface (GUI) design concepts are also introduced.
CMPT 116	Computing I	Math	N/A	Gives the fundamentals of programming, including functions, procedures and arrays. It introduces object-oriented programming and GUI components. Some basic numerical methods and engineering applications are also presented. **For engineers**.
CMPT 117	Computing II	CMPT 116 or CMPT 113	N/A	An emphasis on object-oriented programming. Data structures for the storage and efficient retrieval of information will be studied and analyzed, in particular stacks, queues, linked lists and simple binary trees. **For Engineers**.
CMPT 120	Digital document process	N/A	N/A	This course is intended for students interested in how to effectively use modern computer software, and in learning how computers work. It provides an overview of: computer and software components; networking; computer security; basic and

				advanced document preparation; spreadsheets; and data presentation
CMPT 140	Intro to creative computing	N/A	CMPT 105, 111, 113, 116	Concepts in computing such as algorithms, problem solving, and programming are explored using interactive multimedia systems as the creative focus. Basic skills in problem solving, programming, design and interaction, event-based behaviour, and prototyping are developed.
CMPT 141	Intro to CS	CMPT 105 or CMPT 140, and Math	CMPT 115, 117	This course introduces the basic computer science and computer programming principles of algorithms, abstraction, encapsulation, variables, conditional branching, repetition, functions, recursion, and elementary data structures. These concepts are applied to problem solving applications such as data analysis and visualization, simulation, text processing, and image processing
CMPT 145	Principles of CS	CMPT 141, or 111 and permission of department	CMPT 270	Principles include data structures for efficient storage and retrieval of data, selection of appropriate data structures, algorithmic paradigms for solving difficult problems, and analysis of algorithms' time and space requirements. This course also emphasizes fundamental principles of coding style, testing, and top-down design for writing robust, maintainable software.

C.13 University of Toronto [7]

Course Code	*Course Name*	*Pre-requisites*	*Anti-requisites*	*Course Description*
CSC104H1	Computational thinking	N/A	N/A	Introduction to designing programs for systematically solving problems that crop up in diverse domains. Social and intellectual issues raised by computing. Algorithms, hardware, software, operating systems, the limits of computation.
CSC108H1	Intro to computer programming	N/A	CSC120H1, CSC121H1, CSC148H1	Programming in python, elementary data types, lists, maps. Program structure: control flow, functions, classes, objects, methods. Searching sorting, and complexity. Unit testing. No prior programming required.
CSC120H1	CS for the sciences	N/A	Any SCS course except CSC104H1	Intro to CS for students in other sciences. Emphasis on gaining practical skills.
CSC121H1	CS for statistics	N/A	Any SCS course except CSC104H1	Intro to CS for students planning to use computers for statistical analysis and research.
CSC148H1	Intro to CS	CSC108H1	Second year courses	Abstract data types and data structures for implementing them. Linked data structures. Encapsulation and information-hiding. OOP, recursion, efficiency of programs. Students with experience can skip CSC108H1 with permission from faculty.
CSC165H1	Mathematical Expression and Reasoning for CS	N/A	Second year courses	Introduction to abstraction and rigour. Structuring proofs, general problem solving techniques, and running time analysis of iterative programs. Formal definition of Big-Oh.

C.14 University of Waterloo [56]

Course Code	Course Name	Pre-requisites	Anti-Requisites	Course Description
=CS 105	Intro to computer programming 1	N/A	N/A	For students with no previous programming experience. Students will create interactive visualizations, simple games, image processing effects, and explore fundamental imperative programming concepts. General skills such as coding style, modular design, testing, and debugging are also explained.
CS 106	Into to computer programming 2	CS 106	N/A	Explore more general applications of programming in contexts of interest to visual artists and designers.
CS 115	Intro to computer science 1	N/A	CS 135	This course introduces students to key concepts in the field of computer science using a functional programming language
CS 116	Intro to computer science2	CS 115, CS 135	N/A	This course introduces important issues in computer science and transitions into imperative programming.
CS 135	Designing functional programs	N/A	CS 115, CS 145	For students that prefer a more conceptual treatment of introductory computer science in a simple language that is effective but not commercially relevant. The course is designed to be taken by those with no prior programming experience, those with experience will also find it relevant due to its unusual focus.
CS 136	Elementary algorithm design and data abstraction	CS 116 or CS 135 or CS 145	CS 137, CS 138	This course examines elementary data structures and algorithms using the functional and imperative paradigms of computation, and discusses issues surrounding the effective use of programming languages and its application.
CS 137	Programming principles	N/A	CS 115, 135, 136, 145	Introduces software engineering students to elementary data structures, and to the functional programming paradigm.
CS 138	Intro to data abstraction and implementation	CS 137	CS 116, 135, 136, 145, 146	This course introduces software engineering students to elementary data structures, and to the functional programming paradigm.
CS 145	Design, abstraction and implementation	N/A	CS 135, 137, 138	This course introduces students to key conceptions in the field of computer sciences, prepares students to complete the CS portion of their Math core requirements. Greater depth than CS 135.
CS 146	Elementary algorithm design and data abstraction (advanced)	CS 145	CS 116, 136, 137, 138, 145	Familiarizes students with key concepts in introductory computer science from an imperative perspective and to contrast this with the functional perspective. Greater depth than CS 136.

C.15 Western University *[54]*

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
1101A/B	The internet: behind the curtain	N/A	2210A/B	The technological successes that have led to the Internet's wide adoption for work and social purposes. The ways in which computer technology has led to more compact representation of data, and faster, more reliable and more secure communication. Intended primarily for students not in Computer Science.
1025A/B	CS fundamentals I	N/A	1026A/B	The nature of Computer Science as a discipline; the design and analysis of algorithms and their implementation as modular, reliable, well-documented programs written in an object-oriented language (Java). Intended for students with significant programming experience in at least one high-level block-structured or object-oriented language.
1026A/B	CS fundamentals I	N/A	1025A/B	The nature of Computer Science as a discipline; the design and analysis of algorithms and their implementation as modular, reliable, well-documented programs written in an object-oriented language. Intended for students with little or no background in programming.
1027A/B	CS fundamentals II	1025A/B or 1026A/B	1037A/B	Data organization and manipulation; abstract data types and their implementations in an object-oriented setting (Java); lists, stacks, queues, trees; pointers; recursion; file handling and storage.
1032A/B	Information systems and design	N/A	N/A	Techniques used for determining technological needs of businesses; building and managing systems to meet those needs; development roles of individuals and organizations; planning and management of concepts, personnel and processes; related software tools (spreadsheets, databases). Intended primarily for Management and Organizational Studies students.
1033A/B	Multimedia and communication I	N/A	N/A	This course explores the use of different types of media (e.g., text, images, sound, animation) to convey ideas and facilitate interaction. Topics include the design and use of a range of

				software tools for media creation and editing, covering image, sound, animation and video.
1037	CS fundamentals II	1036A/B	1027A/B	A continuation for Engineering Science 1036A/B. Data organization and manipulation; abstract data types and their implementations in an object-oriented setting (C++); lists, stacks, queues, trees; pointers; recursion; file handling and storage. Intended for students in the Faculty of Engineering.
1046A/B	Approachable apps: and intro to programming using JavaScript	N/A	1025A/B, 1026A/B	Foundations of app development for the web and mobile devices. An introduction to basic programming and scripting concepts, and technologies such as JavaScript, HTML, and CSS, which will be used to create a variety of apps and games. This course is intended for students with no prior programming or computing background.

C.16 Carnegie Mellon University [10]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
15-104	Intro to computing for creative practice	N/A	15-112	An introduction to fundamental computing principles and programming techniques for creative cultural practices, with special consideration to applications in music, design and the visual arts. The course develops skills and understanding of text-based programming in a procedural style, including idioms of sequencing, selection, iteration, and recursion. Topics include data organization (arrays, files, trees), interfaces and abstraction (modular software design, using sensor data and software libraries), basic algorithms (searching and sorting), and computational principles (randomness, concurrency, complexity). Intended for students with little to no prior programming experience.
15-110	Principles of computing	N/A	N/A	Programming constructs: sequencing, selection, iteration, and recursion. Data organization: arrays and lists. Use of abstraction in computing: data representation, computer organization, computer networks, functional decomposition, and application programming interfaces. Use of computational principles in problem-solving: divide and conquer, randomness, and concurrency. Classification of computational problems

				based on complexity, non-computable functions, and using heuristics to find reasonable solutions to complex problems. Social, ethical and legal issues associated with the development of new computational artifacts will also be discussed. **Intended for students with minimal or no computing background.**
15-112	Fundamentals of programming and CS	N/A	N/A	A technical introduction to the fundamentals of programming with an emphasis on producing clear, robust, and reasonably efficient code using top-down design, informal analysis, and effective testing and debugging. Starting from first principles, we will cover a large subset of the Python programming language, including its standard libraries and programming paradigms. We will also target numerous deployment scenarios, including standalone programs, shell scripts, and web-based applications. This course assumes no prior programming experience.
15-121	Intro to data structures	15-112	N/A	A continuation of the process of program design and analysis for students with some prior programming experience (functions, loops, and arrays, not necessarily in Java). The course reinforces object-oriented programming techniques in Java and covers data aggregates, data structures (e.g., linked lists, stacks, queues, trees, and graphs), and an introduction to the analysis of algorithms that operate on those data structures.
15-122	Principles of imperative computation	15-112	N/A	**For students with a basic understanding of programming** (variables, expressions, loops, arrays, functions). Teaches imperative programming and methods for ensuring the correctness of programs. Students will learn the process and concepts needed to go from high-level descriptions of algorithms to correct imperative implementations, with specific application to basic data structures and algorithms.
15-128	Freshman immigration Course	N/A	N/A	**The Freshman Immigration Course is taken by first-semester** Computer Science majors on the Pittsburgh campus. The course is designed to acquaint incoming majors with computer science at CMU. Talks range from historical perspectives in the field to descriptions of the

				cutting-edge research being conducted in the School of Computer Science.
15-129	Freshman immigration II	N/A	N/A	Students will learn strategies and techniques for finding information on the Web more efficiently; learn when to start with a search engine, a subject-oriented directory, or other tools; explore and practice using advanced search syntax for major search engines; experience specialized search engines for images, sound, multimedia, newsgroups, and discussion lists as well as subject-specific search engines.
15-131	Great practical ideas for computer scientists	N/A	N/A	This course is intended to help you learn what you need to know in a friendly, low-stress, high-support way. We will discuss UNIX, LaTeX, debugging and many other essential tools. Freshmen only.
15-150	Principles of functional programming	15-151 and 15-112	N/A	This course is an introduction to programming that is focused on the central concepts of function and type. One major theme is the interplay between inductive types, which are built up incrementally; recursive functions, which compute over inductive types by decomposition; and proof by structural induction, which is used to prove the correctness and time complexity of a recursive function. Another major theme is the role of types in structuring large programs into separate modules, and the integration of imperative programming through the introduction of data types whose values may be altered during computation.
15-151	Mathematical foundations for CS	N/A	N/A	This course is offered to incoming Computer Science freshmen and focuses on the fundamental concepts in Mathematics that are of particular interest to Computer Science such as logic, sets, induction, functions, and combinatorics. These topics are used as a context in which students learn to formalize arguments using the methods of mathematical proof. This course uses experimentation and collaboration as ways to gain better understanding of the material. Open to CS freshmen only.
15-199	Special topics: discovering logic	N/A	N/A	This course has the purpose of introducing first-year Computer Science students to elements of formal logic as well as to the

				historical context in which this discipline developed. It will also help you understand and appreciate how CS came about since Computer Science grew out of logic. The specific knowledge and skills you will learn in is course include: an enhanced ability to research topics, give presentations and write technical prose, some elementary logic, some historical depth into Computer Science, mathematics and logic itself. This course is for freshmen only.

C.17 Harvey Mudd [28]

Course Code	Course Name	Pre-requisites	Anti-requisites	Course description
CS 5	Intro to CS	Permission of instructor	N/A	Students learn general computational problem-solving techniques and gain experience with the design, implementation, testing and documentation of programs in a high-level language. In addition, students learn to design digital devices, understand how computers work, and learn to program a computer in its own machine language. Finally, students are exposed to ideas in computability theory.
CS 5GR	Intro to Biology and Cs	Permission of instructor	N/A	This course introduces fundamental concepts from the core course Computer Science 5 using biology as the context for those computational ideas. Students see both the intellectual and practical connections between these two disciplines and write computer programs to explore biological phenomena. Computer science material includes basic data types and control structures, recursion, dynamic programming, and an introduction to automata and computability. This course fulfills the computer science core requirement at Harvey Mudd College. It does not fulfill the HMC biology core requirement.
CS 42	Principles and practice of CS	Permission of instructor	N/A	Accelerated breadth-first introduction to computer science as a discipline for students who have some programming background. Computational models of functional, object-oriented, and logic programming. Data structures and algorithm analysis. Computer logic and architecture. Grammars and parsing. Regular expressions. Computability. Extensive

				practice constructing applications from principles, using a variety of languages.
CS 60	Principles of CS	CS 5 or permission of instructor	CS 42	Introduction to principles of computer science. Information structures, functional programming, object-oriented programming, grammars, logic, logic programming, correctness, algorithms, complexity analysis, finite-state machines, basic processor architecture, and theoretical limitations.
CS 70	Data structures and program development	CS 42 or CS 60	N/A	Abstract data types including priority queues, dynamic dictionaries, and disjoint sets. Efficient data structures for these data types, including heaps, self-balancing trees, and hash tables. Analysis of data structures including worst-case, average-case, and amortized analysis. Storage allocation and reclamation.

C.18 University of North Carolina at Charlotte [15]

Course Code	Course Name	Pre-requisites	Course description
ITCS 1101	Intro to computer concepts	Math	Overview of computer hardware and software. Primary emphasis on productivity software (word processing, spreadsheet, and graphical presentation)
ITCS 1102	Advanced internet concepts	ITCS 1101	For students who are familiar with office productivity tools and a user of internet tech. Topics include concepts of website design and how to evaluate websites, proper use of synchronous and asynchronous communication tools, issues of copyright and cyber-ethics, using the internet to do research, and publishing via the internet.
ITCS 1200	Freshman seminar	Permission of department	Designed to assist students with the transition to college by acquainting them with the university’s resources and support, exploring opportunities in the IT field, and developing a strong sense of community within the department.

ITCS 1203	Survey of computing	N/A	Intro course that explores the broad field of computing as it applies to daily life. Topics include computers of all sizes from handheld devices to super computers, the role of software from operating systems to applications, the software development process, issues of security and privacy on the internet and the world wide web, and possible fields of study within the broad field of information tech.
ITCS 1301	Intro to the financial services industry	N/A	Overview of the financial services industry, including such areas as the industry components, regulatory considerations and their impact, and relations with other institutions.
ITCS 1610	Computing applications Seminar	Math	Informs no-computing specialist of computing technologies, research, and career opportunities.
ITCS 1712	Intro to CS	Math	Intro to algorithmic problem solving using high level programming languages. Basic programming concepts and data types taught in C++ and Java. Advanced concepts such as pointers, references, and polymorphism are explored.
LBST 2213	Science, technology and society (FemmTech)	N/A	The role of science and technology in society. The appreciation and understanding of science and the public policy issues related to science and technology. Issues such as science vs. pseudo-science, the ethics of science and technology, the methods of the sciences, the importance of major scientific discoveries, and public expectations of the sciences.