



# **Computer Science Accreditation Council (CSAC) Sample of Partial Request for Accreditation**

June 9, 2011

## Introduction

The following document shows how key sections 5 and 6 in the Institutional Questionnaire and Self-Assessment Report may be filled out for a fictitious university that follows very simplified CS program shown on this page. We make no judgment about whether the fictitious program would be accreditable.

Sample completed data is in red, and commentary about completing the form itself is in blue. Material in black comes from the questionnaire unchanged. There is no need to use colours, however, when completing the questionnaire.

The fictitious program is as follows, as it might appear in the University's calendar, which would be separately referenced.

### Year 1

- CS11 Introduction to Programming
- CS12 Data structures and Software Construction
- CS13 Overview of Computer Science
- MA11 Calculus 1
- MA12 Calculus 2
- MA12 Logic and Discrete Math
- EN11 Technical Communication
- 1 Science elective from the set CHM1, PHY1, BIO1
- 1 Business, humanities or social science elective from the set BUS1, ART1, PSY1, POL1
- 1 Free elective

### Year 2

- CS21 Algorithms and Data Structures
- CS22 Introduction to Computer Hardware
- CS23 Introduction Software Engineering
- CS24 Databases
- MA21 Probability and Statistics
- MA22 Advanced Computing Mathematics
- 1 Computer science elective
- 1 Science elective
- 1 Business or humanities elective
- 1 Free elective

### Year 3

- CS31 Databases
- CS32 Professionalism and Ethics in Computing
- CS33 Operating Systems
- 1 Computer science electives
- 3 Non-CS, Non-Math electives, at least one of which must be 2<sup>nd</sup> year or above
- 3 Free electives, at least one of which must be 2<sup>nd</sup> year or above

Year 4

CS41 Project – Part 1

CS42 Project – Part 2

CS43 Theoretical Computer Science

2 Computer science electives one of which must be at the 4<sup>th</sup> year level

2 Non-CS, Non-Math electives, at least one of which must be 3<sup>rd</sup> year or above

3 Free electives, at least one of which must be 3<sup>rd</sup> year or above

## 5 Students

### 5.1 Enrollment and graduates in each program

Note that most instructions for how to complete the questionnaire have been deleted to make the result more concise.

Programs to be considered	
Official Program Name	Program Code*
Bachelor of Science in Computer Science	BCS

Current Enrollment				
Program Code	Year 1	Year 2	Year 3	Year 4
BCS	40	45	38	42

Numbers of graduates in each of the last five years					
Program Code	5 years ago	4 years ago	3 years ago	2 years ago	most recently
BCS	36	27	33	41	44

### 5.2 Admission requirements, promotion requirements and passing averages

What are the requirements to enter into or continue in the programs? Please include explicit references to the University calendar or other submitted materials.

**Answer:**

**Students are required to have completed once science course in their final year of high school, and high school mathematics up to pre-calculus. Students are admitted if they have an average of over 80% in their grade 12 Mathematics, Science and English courses. See <http://cs.fictitiousu.ca/admission>**

If you have articulation agreements in place with non-university educational institutions, describe what policies and processes there are in place to assure the equivalency of courses that are recognized under the articulation agreements.

**Answer**

**There are none.**

### **5.3 Student counseling and advising**

How are students advised about course and career selection?

**Answer:**

**The first year is very constrained, so students are automatically registered in their required courses, and choose their own electives from the limited choices. Students are required to meet a staff academic advisor in February or March of their first year to review their file and obtain advice on options for second year. Students with an average of less than 60% after the first term of first year are also required to meet with a faculty member who will serve as their mentor for the rest of their program. Starting in second year, all students meet with their designated mentor once per academic year if their average is over 70%, or every semester if their average is 70% or less. See <http://cs.fictitiousu.ca/advising>**

**The Career Counselling Service is available for students upon appointment at all times. See <http://careers.fictitiousu.ca>**

### **5.4 Quality indicators**

Approximately what percentage of graduates continue in post-graduate studies?

**Answer:**

**We do not have complete data, but we estimate the number to be about 2 per year, or 5%**

How many graduating students received scholarships for post-graduate studies?

**Answer:**

**One student in each of the last three years received an NSERC PGS**

Provide any other data you have indicating the high and/or continuously improving quality of students. This could include prizes awarded, high levels of job placement, feedback from employers, low attrition rates, high admission averages and high graduation averages.

**Answer:**

**100% of the students we have been able to track received job placements in computer science related jobs over the last 2 years or went to graduate school. 97% did so 3 years ago.**

**Starting two years ago, we conducted a survey of all 35 known employers of our graduates, to determine their satisfaction with our graduates in the first four months of employment. The survey listed 20 topics or general skill areas related to the graduate attributes and asked them to rate our graduates on the following scale 1=Poor level of ability; 2=Adequate ability; 3=Good**

ability; 4=Excellent ability and 5=Ability far beyond expectations. 30 of the employers responded 2 years ago, and 33 responded this past year. The survey was conducted by volunteers from our Industrial Advisory Board.

The following are the results:

Question	Rated ability last year	Rated ability previous year
Ability to program	3.2	3.1
Knowledge of data structures and algorithms	3.4	3.3
Knowledge of a variety of tools, languages and technologies	3.3	3.2
Knowledge of computer hardware	2.8	2.9
Knowledge of relevant mathematics	3.0	3.0
Knowledge of relevant business issues	2.4	2.6
Broad general knowledge	3.7	3.9
Ability to analyse and solve computing-related problems	3.8	3.8
Ability to make good design decisions	3.6	3.5
Ability to communicate in written form	2.4	2.4
Ability to communicate orally	2.7	3.0
Professional and ethical attitude	4.0	3.9
Ability to work in a team	3.9	3.8
Ability to work independently	4.0	4.0
Ability to research and learn new knowledge when needed	4.1	3.9

### 5.5 Graduate attributes:

**A graduate of a computer science, software engineering or interdisciplinary program must be able to:**

**GA1. Demonstrate Knowledge:** Competently apply knowledge in a) software engineering, b) algorithms and data structures, c) systems software, d) computer elements and architectures, e) theoretical foundations of computing, f) discrete mathematics and g) probability and statistics.

**GA2. Analyse and Solve Problems:** Use appropriate knowledge and skills, including background research and experimentation, to identify, investigate, abstract, conceptualize, analyse, and solve complex computing problems, in order to reach substantiated conclusions.

**GA3. Design Software and Systems:** Design and evaluate solutions for complex open-ended computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, as well as economic, cultural, societal, and environmental considerations

**GA4. Use Appropriate Resources:** Create, select, adapt and apply appropriate techniques,

resources, and modern computing tools to complex computing activities, with an understanding of their strengths and limitations.

**GA5. Work Individually and in a Team:** Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings

**GA6. Communicate Effectively.** Communicate with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions

**GA7. Act Professionally.** Act appropriately with respect to ethical, societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and with regard to the consequential responsibilities relevant to professional computing practice.

**GA8. Be Prepared for Life-Long Learning:** Learn new tools, computer languages, technologies, techniques, standards and practices, as well as be able to identify and address their own educational needs in a changing world in ways sufficient to maintain their competence and to allow them to contribute to the advancement of knowledge.

**GA9. Demonstrate Breadth of Knowledge.** Possess knowledge in areas other than computer science and mathematics so as to be able to communicate effectively with professionals in those fields.

If you have adopted a different set of graduate attributes than the CSAC defaults, please describe the reasons for doing so, and a mapping to the CSAC default attributes, justifying any omissions.

**Answer:**

**We have adopted the CSAC Graduate Attributes as above**

## 6 Curriculum

One sample page of the per-course document is found on the next page, followed by a sample of the graduate attributes grid. You would fill these out for all courses.

Per-course information requirements	Details
Course Code(s):	<b>CS21</b>
Course Name:	<b>Algorithms and Data Structures</b>
Program Codes:	<b>All</b>
Names of professors who have most recently taught the course	<b>John Smith, Marie DeLaplante</b>
GA1 Demonstrate Knowledge.	<b>Sorting, searching, hash tables, algorithms on graphs, complexity analysis, computability, grammars and parsing, use of libraries</b>
GA2. Analyse and Solve Problems:	<b>Students are presented with algorithm analysis and design problems in biweekly assignments. There are assignments that require students to a) design and implement an algorithm and/or a data structure; b) modify existing algorithms or code (demonstrating an ability to understand complex algorithms); c) analyse the complexity of algorithms; d) select and use data structures and algorithms from a library. 40% of the grade is based on these individual assignments; 60% of the grade is based on tests and the final exam that test whether students have in fact learned the material.</b>
GA3. Design Software and Systems:	<b>As mentioned in GA2, there are algorithm and data structure design assignments</b>
GA4. Use Appropriate Resources:	<b>Students are taught to use both the Java and C++ STL libraries. There are assignments requiring the use of these. Both are widely used in industry. Although this is not a programming course per se, students are required to follow disciplined procedures as they use these tools, including test-first development and carefully documenting their code.</b>
GA5. Work Individually and in a Team:	<b>All work in this course is individual.</b>
GA6. Communicate Effectively.	<b>Nothing relevant in this course.</b>
GA7. Act Professionally.	<b>There is one hour of discussion regarding patents on algorithms, and students are taught to observe copyrights and license requirements when reusing software.</b>
GA8. Be Prepared for Life-Long Learning	<b>One assignment requires students to search in the IEEE-CS digital library for a specific paper, and to implement the algorithm described in that paper.</b>
GA9. Demonstrate Breadth of Knowledge.	<b>One of the algorithm problems each year is either in bioinformatics or chemistry; another is in financial analysis.</b>

## 6.2 Grid, summarizing how courses contribute to each of GA2 through GA9

Complete the following table, with each blank cell being replaced with:

\*\*\* if the course (or course group) contributes *greatly* to the GA in question;

\*\* if the course or group contributes *substantially* to the GA; and

\* if the course or group contributes in a *minor* way to the GA.

Leave the cell blank if the course does not contribute to the GA at all.

Course code(s)	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9
CS21	***	***	**	** (indiv)		*	**	*

## 6.3 Additional quality indicators

In addition to the per-course information given in Section 6.2, please summarize any other qualitative or quantitative assessment you have done which provides evidence that the graduate attributes have been met and that your curriculum is of high and/or continually improving quality. This may include surveys of students, surveys of employers, special tests given to students, interviews with students, etc.

### Answer:

As mentioned in section 5.4, our survey of employers suggests strong abilities among our students.

We have mapped our curriculum to the ACM/IEEE CS2013 curriculum and updated CS21 to add complexity theory, which had been missing.

In their final-year interview with their faculty mentor, all students are asked to complete a survey where they are asked to list their satisfaction with the each of the courses they have taken. They do the survey online and confidentially, but the mentor tells students how important the survey is for improvement of the program. The curriculum committee reviews the results and each year revamps the course or courses that get the lowest ratings.

Last year this resulted in the restructuring of the Software Engineering course; the previous year

it resulted in adding some topics to the Artificial Intelligence course and deleting other topics. Overall satisfaction in the survey went up from 3.9/5 to 4.0/5 in the last year.

**6.4 Expected minimum numbers of courses in various categories**

Complete the following table to indicate how the *minimum* requirements in each program compare to CSAC’s general expectations. Use one row for each program being considered. In each cell please give the coeds and abbreviations for the required courses (or groups of courses) in your program(s) that fall in each category.

Please note that under outcomes-based accreditation, fulfilling graduate attributes takes precedence over strict counting of numbers of courses, so the guideline numbers given in rows two and three may not need to be fully adhered to in all cases. The material in this table will help assess GA1 (knowledge taught) and GA9 (breadth of knowledge).

<b>Program Code</b>	<b>CS/SE</b>	<b>Math</b>	<b>non-CS / non-Math</b>	<b>Unspecified</b>
<i>Guidelines for CS and SE programs</i>	$\geq 15$	$\geq 5$	$\geq 10$	$\leq 10$
<i>Guidelines for Interdisciplinary programs</i>	$\geq 10$	$\geq 3^1$	10 (at least 5 in each Other Discipline)	At least 3
<b>BCS</b>	<b>17+</b>	<b>5+</b>	<b>10</b>	<b>8</b>

**Additional-comments**

---

<sup>1</sup> In case where mathematics is one of the Other Disciplines the total required courses is 10 (not 13). Courses in column three and four can be double counted.

### 6.5 Coverage of areas of computing

Complete the following table to indicate the codes and abbreviations for the *required* courses taken by students that significantly address each of the identified sub-areas within Computer Science, again using one row per program.

For general CS and SE programs, it is the general expectation that there will be material taught in all of the categories. However for interdisciplinary programs, it may be that only algorithms and data structures taught. As before, fulfilling the graduate attributes is key. The material in this table will help assess GA1 (knowledge taught) and GA9 (breadth, within computing).

Program Code	Software engineering	Algorithms and data structures	Software systems	Computer elements and architecture	Theoretical foundations
BCS	CS23-SE	CS12-DSSC, CS21-Alg	CS33-OS	CS22-HW	CS21-Alg, CS43-Th

#### **Additional-comments**

### 6.6 Special types of knowledge

What knowledge and skills at an advanced level will students learn in each program? (Used to assess GA1)

#### **Answer**

**Students learn advanced knowledge in the areas of operating systems (CS33-OS) and theoretical computer science (CS43). They also have to take 3<sup>rd</sup> and 4<sup>th</sup> year electives in CS. Additionally some of their non-technical and free electives have constraints to be from third year or above.**

How do the programs guarantee students are exposed to multiple programming languages and paradigms? (Used to assess GA1 and GA4)

#### **Answer**

**Students learn Java and C++ in core first and second year courses. They learn C in CS33-OS, and are intruded to functional programming in CS43-Th, in order to explore the practical implications of various theoretical issues.**

How do the programs expose students to new areas of computing? (Used to assess GA1 and GA9)

#### **Answer**

**Ten electives are offered each year at either the third or fourth year level, and students have to choose from among them. Recent courses have included DNA computing, Data mining, and Algorithms in bioinformatics.**

**6.7 Coverage of areas of SE, for SE programs**

For programs to be considered under the **Software Engineering Accreditation** guidelines, please indicate the required courses in the program that cover the following areas of Software Engineering. This is to assess GA1.

N/A

**Additional-comments**

**6.8 Coverage of areas of mathematics**

Complete the following table to indicate the *required* courses taken by students that significantly address each of the identified sub-areas of mathematics, again using one row per program. This is to assess GA1.

It would generally be expected that in CS and SE programs, there is material taught in each of columns 2-7. For interdisciplinary programs Discrete Mathematics and Probability and Statistics would be considered normally essential, combined with some grounding in logic, Boolean algebra and the basics of graph theory.

Program Code	Discrete math	Calculus	Probability and statistics	Logic	Boolean algebra	Graph theory	Other math
BCS	MA12, CS21, CS43	MA11, MA12	MA21-PS	MA12	MA12	MA12, MA22, CS21, CS43	

**Additional-comments**

**6.9 Breadth requirement**

Complete the following table to indicate the *required* courses (or categories of courses) taken by students in each of the broad disciplines outside of computer science and mathematics/statistics, again using one row per program. This is to assess GA9.

Program Code	Science	Engineering	Business	Humanities	Social science
BCS	2	0	0	0	0

Note: For CS and SE programs, a minimum of ten courses is suggested in these areas, with a minimum of three courses in humanities or social science. For interdisciplinary programs a

minimum of three courses are suggested in these areas.

**Additional-comments**

No specific courses in business, humanities, or social sciences are required, but between them 7 are required.

**6.10 Required courses in Other Disciplines for interdisciplinary programs**

For **Interdisciplinary** programs complete the following table to indicate the *required* courses (or categories of courses) taken by students in the Other Discipline(s) of the programs. Again using one row per program. This is to assess GA9.

Program Code	Courses
N/A	

Note: A minimum of 10 courses is required in these areas, with at least 5 in each Other Discipline (two of which must be advanced courses)

**Additional-comments**

**6.11 Additional questions regarding curriculum**

How does the Department manage and review its curriculum?

**Answer**

The department has a curriculum committee chaired by the vice-chair. This meets at least once every semester. Each year the course that received the lowest rating in the student’s survey is reviewed. The whole curriculum is reviewed every three years by the Industrial Advisory Board, and the curriculum committee acts on their recommendations.

Are there other innovative aspects of the programs that deserve special mention?

**Answer**

Two field trips to local computing companies are given in each of the two fourth-year project courses. Students learn about life and technology of the companies visited.