Canadian Information Processing Society

Computer Science Accreditation Council

Accreditation Criteria
for
Undergraduate Software Engineering Programs

2008/2009
Computer Science Accreditation Council

Criteria for Programs in Software Engineering

Abstract

These guidelines are written to provide assistance to faculty and administrators involved in the accreditation of Software Engineering programs within public and private not-for-profit universities. They specify the objectives of accreditation, the various steps in the process, and the essential and highly desirable qualities of accreditable programs. Questions and suggestions for improvements may be sent, either directly or through CIPS National Office, to the Chair of the Computer Science Accreditation Council, who will ensure that they are considered.

Computer Science Accreditation Council (CSAC) Policies

1. The Computer Science Accreditation Council (CSAC) assesses undergraduate programs for accreditation. The institution is responsible for identifying the program(s) which is/are to be assessed. A program is normally identified as such within the Calendar of the institution, and should be characterized by a curriculum that is regarded and promoted as an entity by the institution, and which may be considered independently. A program may contain options and/or electives, and it is the responsibility of the institution to identify all such options and electives within a program to be assessed. In assessing a program for accreditation, CSAC will examine all such options and electives within the program, and will accredit a program only if all such options and electives meet the criteria.

2. To invite institutions to submit programs for accreditation by the Council.

3. CSAC accredits a program only if students have graduated from that program. However, an accreditation visit for a new program may be undertaken during the final academic year of the first graduating class of that program. Accreditation of such a new program is granted only if and when students graduate from the program in the year in which the accreditation visit takes place. The effective date of such accreditation will include the first graduating class. Should the program fail to produce graduates in the academic year of the visit, accreditation will be denied.

4. CSAC will give primary consideration to programs as taken by students who will be graduating within two academic semesters (10 courses) following the accreditation visit. However, if the requirements of a program have been officially changed for students who will be graduating at a later date, then CSAC will also assess whether the new requirements meet the accreditation requirements. CSAC is only able to evaluate the content of courses which have been actually taught.

5. To favour broad programs that will prepare students to take advantage of as many different opportunities as possible, to contribute to the needs of Canadian society both in the near and long term, and to embark on life-long learning.

6. To deny accreditation to programs that omit instruction in a significant portion of a subject in which computer professionals may reasonably be expected to have competence. This policy is intended as a safeguard to the public and should not entail the setting of rigid standards.
7. To avoid rigid standards as a basis for accreditation in order to prevent standardization and conservatism, and to encourage planned experimentation.

8. To assess qualitative as well as quantitative factors in making an accreditation decision. This should be implemented by a visit to the institution by a competent committee, having suitable qualifications.

9. CSAC accredits satisfactory programs for a full term. If the CSAC judges, that there are areas of concern, accreditation may be granted for a three year term. The areas of concern and method by which the CSAC will assess if all concerns have been addressed will be explicitly stated. The assessment methods will include either the review of a report submitted by the institution or a visit by an accreditation team. It is expected that the institution will take action to bring the program into full compliance with the criteria.

10. To revoke accreditation if institutions do not continue to comply with established criteria. If it appears that an accredited program is not in compliance with the criteria, then the institution is so notified. If the response from the institution is not adequate, then the Council will institute procedures to determine the actual status of the program in question prior to making any decision with respect to the revocation of accreditation.

11. To provide means for reconsideration.

12. To publish a list of accredited programs only. Information such as to whether a program not on the accredited list had been under consideration by the Council will not be made available except to the appropriate officials of the institution offering the program.

13. An institution may not simultaneously use the same program title to identify both an accredited program and a non-accredited program.

14. CSAC will consider the review and accreditation of privately funded university programs if the program has satisfied the following conditions:

   a) Received provincial government recognition involving institutional recognition/accreditation and a program specific review recognition/accreditation and continues to meeting the province’s guidelines for continuing the recognition.

   b) Is a baccalaureate degree program.

Note: All programs requesting accreditation will be reviewed by CSAC existing accreditation criteria. No changes or exception will be made to these criteria to accommodate special needs or requirements that exist in privately funded educational institutions.

Introduction

The Computer Science Accreditation Council is an autonomous body established by the Canadian Information Processing Society (CIPS).

The Council has as its objectives:

1. To formulate and maintain high educational standards for Canadian universities offering computer and information science programs, and to assist those institutions in planning and carrying out educational programs.

2. To promote and advance all phases of computer and information science education with the aim of promoting public welfare through the development of better educated computer professionals.

3. To foster a cooperative approach to computer and information science education between
industry, government, and educators to meet the changing needs of society.
The purpose of accreditation is to recognize programs whose graduates will have received an outstanding undergraduate education in software engineering, an education informed by state of the art research, the mathematical underpinnings of computer science, and the needs and applications of industry.

Method of Evaluation

Programs submitted for accreditation will be evaluated on the basis of data submitted by the institution in the form of a questionnaire and other supporting documentation, together with the report of an on-site visit by a carefully selected team representing the Council. The purpose of the site visit is three-fold.

First, the site visit should assess factors beyond those described in the questionnaire. The intellectual atmosphere, the morale of the faculty and the students, the calibre of the staff and the student body, and the character of the work performed are examples of intangible qualitative factors that are difficult to document in a written statement.

Second, the visiting team should help the institution assess its weak as well as its strong points. Third, the team should examine in further detail the material compiled by the institution and relating to:
1. Control and organization of the institution.
2. Education programs offered and degrees conferred.
3. The basis of and requirements for admission of students.
4. Number of students enrolled: a. in the college, faculty or division as a whole, b. in the individual educational programs.
5. Teaching staff and teaching loads.
6. Commitment to and support for research.
7. Resources: a. financial: total budget, non-salary portion of budget and salary scales, b. physical: classrooms, laboratories, equipment and offices, c. support staff: administrative, clerical, laboratory, research and technical, d. library.
8. Curricular content of the program.
9. Actual course selections, as reflected by a sample of anonymous transcripts.
10. Innovative and special features of the program.

Faculty

The heart of any educational program is the faculty. A competent, qualified, and forwardlooking faculty gives an overall scholarly and professionally responsible atmosphere to the operation. An excellent faculty will usually identify and overcome problems in other areas and continue to provide a program worthy of accreditation, but no degree of excellence in other areas can continually offset the handicap presented by poor faculty quality or inadequate numbers of faculty. Thus, the first consideration for a program to be acceptable for accreditation is the presence and future assurance of a continuing critical mass of quality faculty. Educational institutions seeking CSAC accreditation of programs must have allocated the resources necessary to achieve a critical mass of quality faculty who are committed to professionalism, and must be committed to maintaining the allocations required for its continuation.

The proper size of the faculty depends on the enrolment and the program objectives, including amount of sponsored research, direction of graduate research, extension and continuing
education activities, and involvement in professional and technical societies. The number of faculty members must be large enough to provide a broad range of experience and capability and to provide meaningful technical interaction among the faculty members so as to support these interests. The faculty should occupy permanent positions to ensure continuity and stability. Institutions with limited enrolment and resources are encouraged to select and emphasize a smaller number of quality programs rather than to compromise standards by initiating or trying to maintain programs with inadequate faculty support.

To function effectively as teachers, faculty members must devote a significant amount of their time to seeking new understanding through research and scholarship, industrial interaction, instructional innovation, consulting, or other professional development activities. A significant common aspect of these activities is communication of ideas to other practicing professionals, scientists, and engineers outside the home institution. Teaching loads must leave enough time for professional development of the faculty.

Sabbatical leaves are important to faculty development, for they offer the individual an opportunity to develop professionally and allow for visiting faculty. Other evidence of institutional interest in faculty development, such as adequate resources for professional development, should be present.

**Students**

An accredited program must have good students. Student selection and retention standards must be appropriate to the program. When students transfer from other institutions or from a branch campus, standards for evaluation and selection of these students should be clearly enunciated, and should show that these students are of similar quality and have substantially the same knowledge as those students who have taken all their work on the main campus. When computer courses are regularly taken on other campuses, the main campus faculty should be involved with answers to questions of curricular content.

A student advisory system is an important component in a computer educational program. The advisory system should embrace course selection and similar matters, and it should also include career guidance. Various aspects of professionalism and ethics may be dealt with through the guidance system.

Curriculum and career guidance is best handled by well-informed faculty members who are given the time and administrative support for personal interaction with individual students. Both faculty members and advisors should be familiar with accreditation policies and guidelines, professionalism issues, professional certification, ethical codes of conduct (e.g. CIPS Code of Ethics and the CIPS I.S.P.).

The level of guidance needed will be a function of the flexibility of the curriculum in a particular school. Care should be taken by advisors not to assume responsibility of choice which should be exercised by the student.

**Curriculum**

In specifying criteria for achieving accreditation in any discipline, there is a tension between establishing minimum standards and encompassing flexibility within which students can set individual goals and tailor their programs to meet diverse needs. The Computer Science Accreditation Council has adopted the following criteria to strike a balance between these
objectives.

In the following, the unit of instruction is a course. Different institutions define courses and years and programs in different ways. For purposes of clarity in this document, a course is a single semester course, of roughly 12 weeks in length consisting of approximately 36 lecture hours. Thus a four year undergraduate program would consist of roughly forty (40) courses. It is important to note that although the CSAC unit of counting is in terms of a course, a specific program may meet a requirement of a ‘course in a specific subject area’ by material covered in two or more courses. Each course or part thereof may only be counted once in meeting the criteria in this document.

Further, an introductory course is a course for which has no university-level course as a required prerequisite. An intermediate or advanced course has at least one university-level course as a required prerequisite. In many universities, introductory courses are referred to as ‘first year courses’, while an intermediate or advanced course might be referred to as an ‘upper year course’.

It is recognized that institutions assign title to courses in differing ways, and it is also recognized that courses are offered by different academic units within different institutions. In determining the extent to which a course or courses meet(s) (a) requirement(s) specified herein, it is the content of the course(s), as evinced by the course outline(s) and the course materials, which are of significance, and not the course title(s) or the particular academic unit offering the course.

***Note for Quebec institutions***

In Canadian provinces other than Quebec, a student typically obtains his or her degree after sixteen years of study, including grades one through twelve, plus four years of university. In Quebec, a student typically obtains his or her degree after sixteen years. Eleven years of primary and secondary school, two (general curriculum, pre-university) or three (technical curriculum leading to university) years of CEGEP, and three or two years of university.

The criteria below (explained in terms of numbers of courses) are specified assuming a university program of four years (forty courses). In order to satisfy the criteria a program in Quebec may therefore include up to:
* Ten CEGEP courses for students who have obtained a (general curriculum) pre-university CEGEP diploma;
* Fifteen CEGEP courses for students who have obtained a technical CEGEP diploma (leading to a university program).

*** End of note for Quebec institutions***

I. An accredited program must include the equivalent of forty courses of study (not necessarily all at the same institution) and lead to a Baccalaureate degree or equivalent. Furthermore it must require that all students satisfy certain breadth and depth criteria to be eligible for the degree. Within the forty courses, there must be:

* 15 (fifteen) courses in computer science/computer engineering/software engineering
* 5 (five) courses in mathematics/statistics, and
* 10 (ten) courses in subjects other than computer science/computer engineering/mathematics/statistics
The set of courses in each area should exhibit some breath and some of the courses in each area should be at an intermediate or advanced level. This thirty course requirement leaves each student with the equivalent of one year of study to satisfy additional institutional requirements and to accomplish personal objectives.

II. As an adjunct to specific areas of instruction, the program must also include training to develop students’ written and oral communications skills. Not only should students be taught to present information both verbally and in writing, but they should practice collecting information through reading and listening and assembling the information for various audiences. The realization of this requirement in the program can be tailored to suit other institutional goals, but the involvement of the computer science faculty must be evident.

III. Similarly to the program as a whole, the curriculum within computer science must offer a breadth of exposure to different topics, along with a depth of understanding. A. All students in an accredited program must be required to take courses in the following five areas, each of which is illustrated by a range of topics that might be included. Introductory courses in computer programming and computer usage may not be used to satisfy these requirements. This classification of subtopics is not intended to limit the scope of computing nor the areas to be included or excluded at particular institutions, but rather it is illustrative of a range of topics that an accredited program would be expected to provide.

1. Software engineering including software requirements (elicitation, analysis, specification, validation), software design and architecture (architectural structures, styles and patterns, structural vs. behavioral descriptions, strategies and methods), software construction and maintenance (abstraction mechanisms, assertions, debugging, coding style, programming environment and tools, including configuration management), software testing and quality assurance (unit, integration and system testing, black/white box testing techniques, validation and verification, reliability); software engineering management and process (metrics, estimating, planning, scheduling); applications of S.E. to various areas (embedded, real-time or distributed systems, human computer interaction, databases, etc.).

2. Algorithms and data structures, including data structures such as stacks, trees, lists, queues, etc., abstract data types, established solutions to classical problems (e.g., sorting and searching), and analysis of algorithms.

3. Systems software, including operating systems concepts, virtual memory management, management of distributed, parallel, and concurrent processes, transaction processing, logging, security, and computer networking.

4. Computer elements and architectures, including computer organization, digital device and communications technology, logical and physical hardware design.

5. Theoretical foundations of computing, including models of computation, analysis of algorithms, fundamentals of program specification and verification, computational complexity, grammar and automata, etc.

The curriculum must require every student to have completed the equivalent of (at least) three (3) courses covering the various software engineering topics under item 1, and also one (1) course in each of the other four areas listed above. Such a course need not cover the entire area, but should provide students with a reasonably broad introduction to the area.
Furthermore, at least one-third (five courses) of the course requirement in computing must be met by intermediate or advanced courses chosen from at least two different areas of Computer Science.

B. Students graduating from an accredited program should have proficiency in at least one programming language and exposure to a variety of programming languages. Exposure to a variety of programming paradigms – procedural, object-oriented, logical or functional, sequential and concurrent – is also important.

C. Accredited programs must prepare students to meet the computing challenges they will face after graduation, whether they embark on careers immediately or continue their education. Thus, as part of their undergraduate education, students must be well-grounded in state-of-the-art computing practices. An accredited program should expose students to several computing configurations, including varied hardware, operating systems, and programming environments.

D. Computer Science is a rapidly developing and growing subject. At the current time, these newer developments include such areas as multimedia, artificial intelligence expert systems, robotics, data mining, human computer interaction, computer graphics, security, networks, bioinformatics, geographical information systems, and so forth. While it is not to be expected that an accredited program will include material in all of these areas, accredited programs should nonetheless demonstrate that they recognize the rapidly evolving nature of the subject, and should include some of the newer areas of the subject, particularly within the intermediate and advanced courses which they offer to their students.

E. Aspects of professionalism are to be emphasized throughout the curriculum. A specific course or courses in social implications of computing may be offered, but ethical and legal issues surrounding computing, including the social responsibility of programmers and computer users, must be emphasized in courses throughout the program so that students learn that these aspects are part of computing, not merely tangential disciplines.

F. Students should also be made aware of the variety of existing standards of Practice (Practice Guides, Recommended Practices, Practice Standards), including those endorsed by CIPS (e.g., ISO/IEC 12207).

G. A significant component of an accredited program must be practical in nature; students must have direct experience with non-trivial problem-solving, and in the case of software engineering programs, students must have direct experience with the major phases of the software development lifecycle. This component helps to develop students’ creativity in solving open-ended problems through practice in formulating problem statements and specifications, considering alternative solutions, determining feasibility and cost, and communicating the results including detailed systems descriptions. Projects and courses that require teamwork are strongly encouraged.

Whereas practical aspects should be included at all levels of the program, the major portion of the practical component is to be satisfied in advanced courses. Students, individually and as members of teams, must be required to design and implement a system, program, process, or device to achieve stated objectives. The problem solution should include the establishment of objectives and criteria, analysis, synthesis, documentation, implementation, testing, and evaluation. It is desirable that problems include a variety of realistic constraints such as economic
factors, performance thresholds, ergonomics, compliance with standards, interoperability with other systems, and conformance to ethical, professional and legal restrictions.

IV. Computing applications can be found in all human endeavors, so education in any discipline outside computer science has the potential to prepare students in a field of direct relevance to their future livelihood.

The diversity of backgrounds needed by various computing professionals necessitates flexibility within these accreditation requirements. Innovation in establishing institutional requirements or in promoting each individual’s ability to reach personal goals is encouraged. Nevertheless, this section identifies topics that enhance a program’s ability to meet the needs of computer science students. As before, the goal of breadth in exposure must be balanced by the goal of depth in understanding, which can best be achieved through the selection of complementary courses having a common focus.

The minimum requirement of five courses in mathematics must include at least one course in each of the following three areas (i) Discrete Mathematics, (ii) Differential and Integral Calculus, and (iii) Probability and Statistics. Students are expected to receive a solid grounding in Logic, Boolean algebra and graph theory. Other topics of particular relevance to computing should also be offered, including a selection from Linear Algebra, Set Theory and Modern Algebra, Numerical Analysis, Differential Equations, Optimization, and Queuing Theory.

Outside of computing and mathematics courses, students should be encouraged to choose courses such that their programs include:

* five (5) courses in science, engineering (but not computer or software engineering), or business
* five (5) courses in humanities or social science

Topics in physics and electrical engineering are basic to many aspects of computing, and courses in these areas are particularly encouraged. Challenges such as endeavors to map the human genome underline the value of education in other fields of natural science as well, including chemistry, the earth sciences, and the life sciences, especially when integrated with computer studies.

A thorough grounding in business fundamentals is important to prepare students of computer science to contribute to Canadian industry. Relevant courses in this area include particularly accounting, business organization, economics, and auditing, but also include marketing, personnel management, and production management.

Students in computing must also be educated in non-technical disciplines. Through courses in humanities or social science, students will gain understanding of political theories and processes, knowledge of individual and group social interactions, appreciation of cultures and history, sensitivity to the literary and fine arts, and fluency in languages. Not only is this important background for future self-study, but several aspects have direct bearing on particular areas of computing: linguistics is central to natural language processing, cognitive science provides mechanisms to evaluate human-computer interaction, law can be applied to assessing liability of computer professionals, and ethics helps to evaluate social implications of computing.
Resources

An accredited program must have buildings, offices, laboratories, computing facilities and equipment, support staff, fiscal resources that are appropriate for the characteristics of the program that is being undertaken, and evidence to this effect should be presented. The availability of sufficient computing resources and support staff is of vital importance to a computing program. An appropriate variety of computational and network facilities must be readily accessible to all students and faculty, and access should be provided not only during scheduled laboratory class hours but also at other times.

The program must have competent administrative and technical support and services. Salary budgets must be consistent with the faculty size and student enrolment. Current expense budgets must allow reasonable amounts of travel and supplies. Computer budgets must allow students and faculty enough computer time that they use it as an effective learning aid. There must be adequate access to library materials and other intellectual resources to support the program. The collections must be maintained and refreshed so as to remain current, and there must be a breadth of materials included. Electronic networking sufficient to provide students and faculty access to external resources is also important.

Version: September 2005
Revised on: December 22, 2008
Computer Science Accreditation Council

Accreditation Criteria
for
Undergraduate Software Engineering Programs

Canadian Information Processing Society
5090 Explorer Drive
Unit 801
Mississauga, Ontario L4W 5A6
Tel: (905) 602-1370
Info@cips.ca
accreditation@cips.ca