Understanding the New ABET Computer Science Criteria

Michael J. Oudshoorn
Northwest Missouri State University
Maryville, Missouri
oudshoorn@nwmissouri.edu

Stan Thomas
Wake Forest University
Winston-Salem, North Carolina
sjt@wfu.edu

Rajendra K. Raj
Rochester institute of Technology
Rochester, New York
rajendra.k.raj@rit.edu

Allen Parrish
United States Naval Academy
Annapolis, Maryland
aparrish@usna.edu

ABSTRACT

The Computing Accreditation Commission of ABET accredits over 300 Computer Science programs worldwide under the Computer Science program criteria. In the past few years, the Commission has engaged the computing community to modify and update these criteria, and has now signed off on an updated set of program criteria that take into account factors such as the CS2013 curricular guidelines, impact on currently accredited programs, structural changes to clarify the criteria, and the reduction of the assessment burden. These recent changes to the Computer Science program criteria primarily impact student outcomes (what program graduates are expected to know and be able to do by graduation) and curriculum. Those changes will impact computer science programs in a variety of ways and degrees—some programs will be impacted significantly while others will be affected in a relatively minor way. This paper examines the changes that will take effect starting in the 2019-20 accreditation cycle, covering the rationale for those changes and exploring the likely impact on computer science programs that are currently accredited or seeking accreditation in the near future.

CCS CONCEPTS

• **Social and professional topics → Accreditation**; **Computer science education**;

KEYWORDS

Computer science degree programs; computing programs; program accreditation

ACM Reference Format:
Michael J. Oudshoorn, Stan Thomas, Rajendra K. Raj, and Allen Parrish. 2018. Understanding the New ABET Computer Science Criteria. In *Proceedings of SIGCSE '18: The 49th ACM Technical Symposium on Computer Science Education, Baltimore, MD, USA, February 21–24, 2018 (SIGCSE '18),* 6 pages. https://doi.org/10.1145/3159450.3159534

SIGCSE '18, February 21–24, 2018, Baltimore, MD, USA

ACM ISBN 978-1-4503-5103-4/18/02…$15.00
https://doi.org/10.1145/3159450.3159534

1 INTRODUCTION

The Council for Higher Education Accreditation (CHEA) describes higher education accreditation as "... a collegial process based on self and peer assessment for public accountability and improvement of academic quality" and states "Accreditation involves judgments of quality and effectiveness of an institution/program against a set of expectations (standards, criteria)" [5]. With respect to the accreditation of computing programs by ABET, the expectations of programs are documented in the ABET Criteria for Accrediting Computing Programs (Criteria) [2, 3].

Accreditation of Computer Science programs has been ongoing since visits were conducted by the Computer Sciences Accreditation Board (CSAB) in the 1985-86 academic year. CSAB continues to play a role in computing accreditation as one of thirty-three cooperating member societies of ABET [6]. The integration of CSAB into ABET occurred in 2001 with the formation of a new ABET commission to focus on computing accreditation, the Computing Accreditation Commission (CAC). CSAB continues to play a key role in the formulation of accreditation criteria for computer science, information systems, information technology and software engineering programs, and recruits and trains program evaluators. The CAC of ABET conducts program reviews against the relevant criteria and makes accreditation decisions. Presently, CAC accredits 461 programs at 354 institutions around the globe, including 338 undergraduate computer science programs, 58 information systems programs, and 50 information technology programs [1].

1.1 The Computer Science Criteria

The CAC Criteria for Accrediting Computing Programs [2, 3] are composed of the General Criteria that apply to all computing programs, and Program Criteria specific to computer science, information systems, or information technology, as determined by the program name. This paper focuses on the recently approved revisions to those portions of the CAC Criteria that apply to computer science programs, which we refer to as the computer science criteria. We refer to the revised criteria as the 2019 criteria, as they will required in the 2019-20 cycle, and the current criteria as the 2017 criteria.

We review the rationale for the 2019 criteria revisions and the potential impact on computer science programs that are currently accredited or seeking accreditation in the near future.

2 FACTORS GUIDING CRITERIA REVISIONS

The last major change to the accreditation criteria for computer science occurred around 2006 when CAC adopted a version of Engineering Criteria 2000 (EC 2000) [8], and transitioned from a standards-based approach to an outcomes-based approach. This significant change helped to shift the foundation of accreditation from *what is taught* by a program to *what is learned* by students by the time they graduate from the program. Since then, although computing infrastructure and technologies have evolved significantly, there have not been significant changes in the CAC accreditation criteria, thus creating a groundswell for criteria revisions.

2.1 CS 2013

The Computer Science Curricula 2013 report (CS2013) [4], published by ACM and the IEEE-Computer Society in December 2013, is likely to impact computer science curricula for many years. The report, developed over a four year period with broad community input, represents a comprehensive revision of previous curriculum recommendations and includes a redefined body of knowledge, desirable characteristics of graduates, and many exemplars of courses and programs from a variety of institutional contexts. Consideration of revisions to the CAC computer science criteria was informed by the body of knowledge and characteristics of graduates identified in CS2013.

2.2 Student Outcomes

Criterion 3 (Student Outcomes) of the 2017 computing criteria allows programs to identify their own student outcomes based on the objectives of the program and the expectations of external constituencies. Criterion 3 also defines eleven program characteristics, which are shown in Table 1. Often referred to simply as (a)–(k), these characteristics must be enabled but—unlike student outcomes—not necessarily assessed and integrated with the program's continuous improvement process. This distinction between characteristics and outcomes has been confusing to both programs and evaluators over the years.

The 2019 criteria resolve this structural ambiguity, but more importantly, they institute a philosophical change by specifying required outcomes. Rather than allowing programs to select outcomes based on goals and expectations of external constituencies, the criteria now specify required outcomes for *all* computer science programs. At the same time, a program is still permitted to articulate additional outcomes.

2.3 Assessment

As described previously, computer science programs which are accredited under the 2017 criteria can currently elect to adopt characteristics (a)–(k) as the program's student outcomes. If they do so, then the program must assess at least eleven outcomes. There was a desire to reduce this assessment burden to focus the assessment process toward more meaningful continuous improvement. In addition, the use of common outcomes is intended for programs to be able to build common and consistent approaches and tools for assessment, thus leading to the sharing of assessment artifacts, helping to reduce assessment cost.

Table 1: 2017 Criteria: Student Characteristics [2]

The program must enable students to attain, by the time of graduation:	
a.	An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.
b.	An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c.	An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
d.	An ability to function effectively on teams to accomplish a common goal.
e.	An understanding of professional, ethical, legal, security and social issues and responsibilities.
f.	An ability to communicate effectively with a range of audiences.
g.	An ability to analyze the local and global impact of computing on individuals, organizations, and society.
h.	Recognition of the need for and an ability to engage in continuing professional development
i.	An ability to use current techniques, skills, and tools necessary for computing practice.
j.	An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k.	An ability to apply design and development principles in the construction of software systems of varying complexity.

2.4 Other Factors

While a primary goal was to reflect the evolving discipline of computer science, the 2019 revisions to the criteria attempted to maintain backward compatibility with existing criteria, to increase consistency with pertinent engineering criteria, to allow for flexibility in program design, and not to unduly increase requirements without providing trade-offs. An increased degree of consistency between the computer science criteria and the engineering criteria is desirable as many computer science programs are housed within a College of Engineering, and increased consistency helps reduces confusion and workload as each program within the college prepares its self-study report.

3 CHANGE PROCESS

3.1 Committee formation

The change process was managed by a Criteria Committee made up of ABET CAC members and CSAB-appointed volunteers. The committee worked at two levels: (a) an Executive Committee that provided overall management and leadership of the process; and (b) individual subcommittees for each of the disciplinary areas, including computer science, information systems and information technology—as well as a subcommittee for cybersecurity, which was simultaneously developing its own initial draft program criteria for cybersecurity [7].

Revisions were debated within the subcommittees and then submitted to the Executive Committee for further consideration and synthesis across all disciplines. The proposal from the Criteria Committee was approved by the Computing Accreditation Commission in July 2017 and subsequently obtained final ABET approval in

Table 2: 2019 Criteria: Student Outcomes [3]

	Graduates of the program will have an ability to:
1.	Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. *General Criteria*
2.	Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. *General Criteria*
3.	Communicate effectively in a variety of professional contexts. *General Criteria*
4	Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. *General Criteria*
5.	Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. *General Criteria*
6.	Apply computer science theory and software development fundamentals to produce computing-based solutions. *Computer Science Program Criteria*

October 2017 for use in evaluations scheduled for the 2019-20 accreditation cycle. Programs may also opt to use the 2019 criteria in the 2018-19 accreditation cycle.

3.2 Feedback regularly sought

The changes leading to the 2019 criteria were developed over a five-year period. This period culminated with a one-year review-and-comment period. During this review-and-comment period, the criteria were posted on the ABET Website and feedback was obtained from the community, both via a web survey and via in-person comments obtained during presentations at conferences such as ACM SIGCSE, ACM SIGITE, IEEE Frontiers in Education, as well as the annual ABET Symposium. A large number of useful comments were obtained during this process. Each comment was analyzed by the appropriate subcommittee and resulted in a recommendation for disposition of that comment, which involved either changing the criteria or leaving the criteria unchanged. Each disposition was documented for internal process traceability.

4 IMPACT OF CRITERIA CHANGES ON COMPUTER SCIENCE PROGRAMS

Of the eight criteria used by ABET to accredit all programs, the CAC criteria changes were limited to only two of them: Student Outcomes and Curriculum. These two, however, are the ones most closely tied to each discipline representing what the students learn and what curricular areas are taught by each accredited program.[1]

Under the Student Outcomes of the 2019 criteria, computer science programs will be required to satisfy six outcomes (five from the General Criteria, and one from the Computer Science Program Criteria) [3]. These outcomes have been consolidated and are shown in Table 2.

The 2019 changes to the Curriculum criterion have been made in both the General Criteria and in the Program Criteria. For ease of understanding, the revised Curriculum section under General Criteria and the Computer Science Program Criteria have been consolidated in Table 3.

Table 3: 2019 Criteria: Curriculum [3]

General Criteria—Curriculum:
The program's requirements must be consistent with its program educational objectives and designed in such a way that each of the student outcomes can be attained. The curriculum must combine technical, professional, and general education components to prepare students for a career, further study, and lifelong professional development in the computing discipline associated with the program.

The curriculum requirements specify topics, but do not prescribe specific courses. The program must include mathematics appropriate to the discipline and at least 30 semester credit hours (or equivalent) of up-to-date coverage of fundamental and advanced computing topics that provide both breadth and depth.

The computing topics must include:

1. Techniques, skills, and tools necessary for computing practice.
2. Principles and practices for secure computing.
3. Local and global impacts of computing solutions on individuals, organizations, and society.

Computer Science Program Criteria—Curriculum:
The curriculum requirements specify topics, but do not prescribe specific courses. These requirements are:

a. Computer science: At least 40 semester credit hours (or equivalent) that must include:
   1. Substantial coverage of algorithms and complexity, computer science theory, concepts of programming languages, and software development.
   2. Substantial coverage of at least one general-purpose programming language.
   3. Exposure to computer architecture and organization, information management, networking and communication, operating systems, and parallel and distributed computing.
   4. The study of computing-based systems at varying levels of abstraction.
   5. A major project that requires integration and application of knowledge and skills acquired in earlier course work.

b. Mathematics: At least 15 semester credit hours (or equivalent) that include discrete mathematics and must have mathematical rigor at least equivalent to introductory calculus. The additional mathematics might include course work in areas such as calculus, linear algebra, numerical methods, probability, statistics, or number theory.

c. Science: At least six semester credit hours (or equivalent) in natural science course work intended for science and engineering majors. This course work must develop an understanding of the scientific method and must include laboratory work.

Most computer science programs will be impacted in some way by the criteria revisions. Changes to Criterion 3, Student Outcomes, will have a more profound impact on some programs than others, depending on the program's current student outcomes. Changes to

[1] It should be noted that although the CAC criteria obviously are applicable to other computing disciplines, this paper focuses on the criteria as applicable to computer science programs.

Criterion 5, Curriculum impact all programs equally. These issues are examined in the remainder of this section.

4.1 Student Outcomes

The 2019 criteria specify six required Student Outcomes (five in the General Criteria for all programs, and one additional outcome for Computer Science), as shown in Table 2. In contrast, the 2017 criteria did not require any outcomes, but instead specified a collection of characteristics, (a)–(k), to be enabled by all programs; these characteristics are shown in Table 1.

Showing that a characteristic is enabled is a weaker standard than showing to what extent a student outcome is being met. Specifically, data collection and evaluation, and documentation of the consideration of that information to enact program improvements is necessary for student outcomes, whereas enabling a characteristic can be demonstrated by showing that students have an exposure or opportunity to gain that skill. That is, these characteristics represent topics that need to be covered in the curriculum, not outcomes that need to be assessed.

However, under the 2017 criteria, programs had the flexibility to create their own student outcomes and assess the level to which these outcomes were being realized. Programs chose one of two approaches to devise their student outcomes:

(1) Adopt the eleven characteristics, (a)–(k), as their own student outcomes, or
(2) Create their own student outcomes different from the (a)–(k).

Programs that took the first approach, i.e., elected to adopt the eleven student characteristics (a)–(k) as their outcomes, will be minimally impacted by the new six required outcomes, since the previous eleven characteristics served as the basis for creating the six outcomes in the 2019 criteria. While the wording of the outcomes differs from the previous eleven characteristics, the content is substantially similar in terms of assessment and evaluation. Programs treating the characteristics as outcomes should already be collecting relevant assessment data. The only change that will be required is potentially converting from eleven to six substantially similar outcomes. For example, 2019 Criteria student outcomes 2, 3, 4 and 5 correspond fairly well to 2017 Criteria student characteristics c, f, e, and d respectively, while outcome 1 covers both characteristics a and b, and outcome 6 more or less covers characteristics j and k.

Programs that took the second approach—elected to identify student outcomes different from the eleven characteristics—often did so to clearly identify differentiating aspects of their programs as compared to their competitors. These programs are not being asked to abandon their outcomes and the assessment data that they have gathered in the past. However, under the 2019 criteria, these programs will need to support the assessment of the six newly required outcomes, in addition to any other outcomes that they want to retain to support continuous improvement. Note that the 2019 criteria permit additional outcomes beyond the required six.

As an example, Table 4 illustrates student outcomes that a program might have selected under the 2017 Criteria [2]. In this example, mapping to the 2019 criteria's student outcomes will require careful consideration. Assessment data currently collected for this program's student outcomes A, B and C would need to be re-factored to allow for a correspondence with the 2019 criteria's student outcomes 1, 2 and 3. However, assessment data previously collected relating to outcomes D, E and F would likely be applicable to the 2019 criteria's student outcomes 5, 3 and 4 respectively.

Table 4: An Example of Program-defined Student Outcomes under the 2017 Criteria [2]

By the time of graduation, students will have the ability to:	
A.	Apply the theory and principles of computer science.
B.	Demonstrate fluency in several programming languages, environments, and tools.
C.	Demonstrate knowledge of computer organization, operating systems, and networks.
D.	Apply computing skills and work effectively in teams in industry or research.
E.	Prepare technical documents and make effective oral presentations.
F.	Analyze and apply legal and ethical issues involving computing in society.

4.2 Computing Curriculum

CS2013 [4] de-emphasizes some aspects of computing such as computer architecture and places grater emphasis on other areas such as information management. It also introduces a number of topics that were not covered in previous curriculum recommendations. These include computer security and information assurance, which is now pervasive throughout the CS2013 recommended curriculum. The 2019 criteria does not increase the minimum amount of computer science (40 credits or $1\frac{1}{3}$ years) that a student is exposed to, but does allow for more computing content to be added if a program desires, through reduction of the mathematics and science credits that a student must complete.

In line with the changes in CS2013, the General Criteria requires exposure to principles and practices of secure computing, as appropriate for the computer science discipline the Computer Science Program Criteria requires exposure to computer architecture and organization, information management, networking and communications, operating systems, and parallel and distributed computing. Several of these are new topics that were not previously specified or required. Care has been taken to explicitly point out that these are not necessarily courses that should be introduced into programs, but rather these are topics that ought to be covered somewhere in the program requirements. For example, some topics may be covered in a relatively small number of lectures embedded within an existing course.

The 2017 criteria require coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture. In comparison, the 2019 criteria requires substantial coverage of several core computer science topics including algorithms and complexity, computer science theory, concepts of programming languages, and software development. These are regarded as fundamental areas of computer science and a knowledge set that differentiates a computer science graduate from a graduate of an information systems, information technology, or cybersecurity program.

The 2019 criteria states that there must be "Substantial coverage of at least one general-purpose programming language". This is a change from "in-depth" coverage of a "high-level language". Most programs are unlikely to be affected by this change which is intended to ensure all students gain proficiency in at least one general-purpose language. The aim is to prevent programs from claiming that they meet this criteria by only requiring a specialized language such as a markup language or an application specific language.

Under the 2019 criteria, students are expected to "study computing-based systems at varying levels of abstraction". Many programs will be unaffected by this change as they already examine computing systems in this way. This can be satisfied in many ways and could include overage of the OSI 7-layer network model, design patterns, ER-models, etc. The goal is for students to be able to abstract away details in order to talk about high-level concepts or to be able to drill down to the appropriate level to identify and solve problems in multi-level systems.

Finally, students must complete a major project that requires integration and application of knowledge and skills acquired in earlier coursework. This is not necessarily intended to be a capstone experience, although it could be one way of meeting the requirement. The intent is that students assimilate and synthesize the knowledge they have gained in their studies and deliver a project that is substantive in nature. This would not typically be a weekly assignment, or maybe even one that a student completes over a 3-week period. Apart from the need for the project to integrate and apply knowledge, nothing is said about the nature of the project to provide programs with flexibility. As this is a significant change, several accredited programs will be impacted unless each develops an approach to meeting this requirement in an appropriate manner.

4.3 Mathematics and Science

The 2017 criteria stated that each "program must include mathematics appropriate to the discipline beyond the pre-calculus level". In addition, the Computer Science Program Criteria stated that there had to be a minimum of one year of mathematics and science of which at least half a year must be mathematics. The mathematics requirement was further elaborated such that it must include discrete mathematics and might consist of other courses such as calculus, linear algebra, and others. The science requirement stated that the courses had to develop an understanding of the scientific method, include exposure to laboratory work, and be classes suitable form science and engineering majors.

There was confusion as to what "beyond pre-calculus" meant and many institutions believed that it was necessary to include pre-calculus as a pre-requisite for all mathematics courses. The list of exemplar courses was also taken somewhat literally by some institutions who clung to other criteria which required calculus courses. There was also confusion as to what a year of mathematics and science meant, with some new programs inquiring if it merely meant two semesters which included a mathematics and science course.

CS2013 does not specify mathematics and science requirements but focuses on the computer science content of programs. Mathematics and science requirements were the topic of significant debate within the criteria committee and at various venues and forums where feedback was sought. Significant feedback suggested that the mathematics and science requirement be reduced or even eliminated. However, the majority still believed that half a year of mathematics was not only appropriate, but necessary, to properly prepare students. A vocal majority also believed that merely one science course, permitted in the previous criteria, was inadequate. Feedback from industry participants strongly supported the need for mathematics and science in many jobs and the belief that students would be poorly served by reducing these requirements.

To address the concerns and confusion, the revised criteria discusses curriculum needs in terms of semester credit hours. The mathematics requirements still require discrete mathematics but indicate that other mathematics courses must have mathematical rigor at least equivalent to introductory calculus. The intent is to signal to programs that calculus is not necessarily required if it does not support the program's objectives as well as other mathematics courses. The list of exemplar courses still remains because many felt that it was necessary to provide some advice to programs, but it is intended to be a list of topics, not names of courses, that might suitably support many currently accredited programs.

The science requirement now explicitly states that there must be at least six credits of natural science course work, which must include laboratory work, and be intended for science and engineering majors. The explicit mention of natural science was to make it clear that course in areas such as the social sciences would not satisfy the criteria.

The impact on programs is that the minimum amount of mathematics and science necessary has decreased from 30 to 21 credits, but the minimum number of science credits is specified as six. Previously, a program with three credits of science and 27 credits of mathematics beyond pre-calculus would have satisfied the criteria. Hence, a potential impact is that some programs may need to increase their science requirement in the degree. There is no reduction in the minimum mathematics requirement, but it is hoped that programs see it as an opportunity to include mathematics courses that best support the program rather than simply requiring traditional courses such as calculus.

5 CONCLUSIONS

The development of accreditation criteria is a challenging process seeking the consensus of a large community with a diversity of opinions. The 2019 ABET computer science criteria ensure that several topics newly identified in CS2013 are now factored into the accreditation process. These criteria also provide a common set of student outcomes as opposed to forcing programs to design their own. These generic student outcomes provide a basis for building a comprehensive continuous improvement process, without being too prescriptive and without forcing programs to go through a potentially elaborate process of reinventing the wheel when developing their own outcomes.

The computer science criteria will continue to evolve in the future, as the discipline matures and comes to closure on what it means to be a BS Computer Science graduate. For example, more work is needed to better capture the role of mathematics in computer science education, an area where additional consideration is needed in the community *writ large*. In addition, having a single set of Student Outcomes for all programs, several of which are based on non-technical capabilities related to team and communication skills, calls for further analysis and consideration.

ACKNOWLEDGMENTS

The following people served on the ABET Computer Science criteria subcommittee in 2016/17: Michael Oudshoorn (co-chair), Stan Thomas (co-chair), Maria Altebarmakian, David Bover, Deepak Garg, Cary Laxer, George Pothering, Gerald Thomas, and Joe Turner. The following people served on the ABET Criteria Executive Committee in 2016/17: Allen Parrish (co-chair), Rajendra Raj (co-chair), Jean Blair, Andrew Borchers, David Cordes, Joseph J. Ekstrom, Scott Murray, Michael Oudshoorn, Srini Ramaswamy, Mark Sebern, Ed Sobiesk, Judy Solano, Stan Thomas, and Heikki Topi. In addition, thanks are expressed to all those who provided input and feedback into the criteria proposal as it evolved over the past three years.

REFERENCES

[1] ABET, Inc. 2017. ABET Webpage. (2017). http://abet.org Accessed: November 14, 2017.

[2] ABET, Inc. 2017. Criteria for Accrediting Computing Programs. Effective for Review During the 2017-18 Accreditation Cycle. (2017). http://www.abet.org/wp-content/uploads/2016/12/C001-17-18-CAC-Criteria-10-29-16-1.pdf Accessed: November 14, 2017.

[3] ABET, Inc. 2017. Criteria for Accrediting Computing Programs, Effective for Review During the 2019-20 Accreditation Cycle. (2017). http://www.abet.org/wp-content/uploads/2017/11/CAC-Criteria-Version-2.0-2018-19.pdf Accessed: November 14, 2017.

[4] ACM/IEEE-CS Joint Task Force on Computing Curricula. 2013. *Computer Science Curricula 2013*. Technical Report. ACM Press and IEEE Computer Society Press. https://doi.org/10.1145/2534860 Accessed: November 14, 2017.

[5] Council for Higher Education Accreditation. 2017. What Is Accreditation? (2017). http://www.chea.org/4DCGI/cms/print.html?Action=CMS_Document&DocID=308&MenuKey=main Accessed: November 14, 2017.

[6] CSAB, Inc. 2017. CSAB Webpage. (2017). http://csab.org Accessed: November 14, 2017.

[7] Joseph J. Ekstrom, Barry M. Lunt, Allen Parrish, Rajendra K. Raj, and Edward Sobiesk. 2017. Information Technology as a Cyber Science. In *Proceedings of the 37th SIGITE technical symposium on Information Technology education (SIGITE '17)*. ACM, New York, 84–89.

[8] Lisa R. Lattuca, Patrick T. Terenzini, and J. Fredricks Volkwein. 2015. Engineering Change: A Study of the Impact of EC2000. (2015). http://www.abet.org/wp-content/uploads/2015/04/EngineeringChange-executive-summary.pdf Accessed: November 14, 2017.