CE2016: Updated Curricular Guidelines for Computer Engineering

Victor Nelson
Department of Electrical and Computer Engineering
Auburn University
Auburn, AL, USA

Eric Durant
Electrical Engineering and Computer Science Department
Milwaukee School of Engineering
Milwaukee, WI, USA

John Impagliazzo
Computer Science Department
Hofstra University
Hempstead, NY, USA

Joseph L. A. Hughes
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, USA

***Abstract*—The report, *Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering (CE2016)*, developed by the Association for Computing Machinery and the IEEE Computer Society, was released in December of 2016. This is one volume of a series of reports covering curricula for a variety of computing fields; it is a significant update of the previous version, CE2004. This paper discusses significant aspects of CE2016, with a focus on how the report might be used in reviewing, updating, and creating computer engineering programs.**

Keywords—computer engineering, curriculum

I. Introduction

The field of computer engineering has seen rapid advances over the past decade and is expected to continue to advance even more rapidly soon. To help in the design of forward-looking curricula to prepare graduates of computer engineering programs for practice, the report titled, *Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering (CE2016)*, developed by the Association for Computing Machinery (ACM) and the IEEE Computer Society, was released in December of 2016 [1]. This report draws upon the 2004 published curricular report in computer engineering titled, *Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*, also known as CE2004 [2]. This report also draws upon recent efforts in computing curricula developed by the ACM, the IEEE Computer Society, and the Association for Information Systems (AIS). These efforts resulted in published curricula recommendations in computer science (ACM/IEEECS, 2013), information systems (ACM/AIS, 2010), information technology (ACM/IEEECS, 2008), and software engineering (ACM/IEEECS, 2015). These reports are available online at the ACM website [3].

CE2004 set out the technical, communication, and professional competencies important at the time. It discussed the state of the field, and the role of the laboratory, engineering tools, teamwork, and lifelong learning in a computer engineer's education. Many computer engineering programs today are heavily influenced by this work. To keep up with rapid advances in computer engineering the societies convened a review taskforce in 2011 to consider whether and how to update CE2004. That taskforce surveyed society members interested in computer engineering and received nearly 300 responses. This input identified the following new and emerging areas that the community believed should be part of a modern computer engineering graduate's knowledge.

- Information security
- Parallel thinking and algorithms
- Concurrent hardware and software design
- Design for testability in digital systems
- Modern embedded system processor types: CPUs, DSPs, GPUs, SOCs, FPGAs
- FPGA development tools
- Mobile embedded systems
- Power-aware software engineering
- Modern development processes including agile methods

Guided by this initial input, the taskforce, which evolved into a steering committee of eleven members listed in Table I, , including one representative from industry, incorporated this information into draft revisions of the guidelines, which were presented, debated, and refined at presentations and workshops at several ACM and IEEE conferences around the world. A key contribution of CE2016 is, thus, ensuring that the intellectual framework for the discipline remains current and aligned with evolving professional practice.

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

TABLE I. CE2016 STEERING COMMITTEE

Member	*Affiliation*
Susan Conry	Clarkson University
Eric Durant	Milwaukee School of Engineering
Lorraine Herger	IBM
John Impagliazzo	Hofstra University
Herman Lam	University of Florida
Joseph Hughes	Georgia Institute of Technology
Weidong Liu	Tsinghua University
Junlin Lu	Peking University
Andrew McGettrick	University of Strathclyde
Victor Nelson	Auburn University
Bob Reese	Mississippi State University

The CE2016 guidelines are meant to *guide* programs and present accumulated knowledge for their consideration as they seek to continually improve. The guidelines are also meant to help new programs as they invent themselves. But, the guidelines are not a mandate. They supplement the vision of educational institutions, industry, and various accrediting bodies in supporting innovation and quality in academic programs. University faculty can use the CE2016 guidelines to design new programs or to see how their programs align with the expert consensus view of the field and then consider whether updates would better serve their constituents. Employers can use them to gain a deep look at the abilities likely to be possessed by newly-graduated computer engineers. Potential students might use them to see how the field aligns with their interests.

II. COMPUTER ENGINEERING AS A DISCIPLINE

To provide context for curriculum design, CE2016 defines computer engineering as a discipline, discussing how the field has evolved, and discussing characteristics of computer engineering graduates, to distinguish computer engineering from computer science, electrical engineering, software engineering, and other disciplines. As an academic field, computer engineering encompasses the broad areas of electrical or electronics engineering and computer science. CE2016 defines this discipline as follows.

> *Computer engineering is a discipline that embodies the science and technology of design, construction, implementation, and maintenance of software and hardware components of modern computing systems and computer-controlled equipment.*

This unique combination prepares students for careers that deal with computer systems from their design through their implementation. Computing systems are components of a wide range of products such as fuel injection systems in vehicles, medical devices such as X-ray machines, communication devices such as smart phones, and household devices such as alarm systems and washing machines. Designing computing systems and computing components for products, developing and testing their prototypes, and implementing them to market are examples of what computer engineers typically do.

The report describes the expected background, knowledge, and skills employers expect to see from graduates of computer engineering programs. These include the ability to design computer systems, the realization of the importance of practicing as professionals, and having the breadth and depth of knowledge expected of a practicing engineer. It also discusses ways in which programs in computer engineering may have to stand up to the scrutiny of validation and accreditation by government or private agencies.

III. THE COMPUTER ENGINEERING BODY OF KNOWLEDGE

The foundation for CE2016 is a fundamental body of knowledge from which an institution could develop or modify a curriculum to fit its needs. This body of knowledge (BoK) contains broad knowledge areas that are applicable to all computer engineering programs worldwide. Changes within this rapidly changing field since the previous report and projections for the next decade prompted significant revisions to the BoK for CE2016.

The BoK has a three-level hierarchical structure. The highest level is the *knowledge area*, which represents a particular disciplinary subfield of computer engineering. The twelve knowledge areas of CE2016, listed in Table II, each contain an *area scope* that describes its context and is then broken down into smaller divisions called *knowledge units* (KUs), which represent individual themes within an area. To capture the sense of what students should learn in connection with each knowledge unit, CE2016 uses *learning outcomes* to describe each knowledge unit. The emphasis on *learning* is important. Taxonomies of verbs such as “define” or “evaluate” are useful to describe the expected depth of learning. Levels of learning range from basic abilities, such as reciting definitions, to advanced abilities, such as engaging in synthesis and evaluation. The verbs used to describe learning outcomes in KUs were influenced by Bloom’s taxonomy [4]. Hence, *learning outcomes* provide a mechanism for describing not just knowledge and relevant practical skills, but also personal and transferable skills. They describe what we expect a student to know or be able to do by the time of graduation. We can infer the minimal desired depth of coverage associated with each knowledge unit from the language used to express the learning outcomes.

One of the goals in creating CE2016 was to keep the required component of the BoK as small as possible. This allows computer engineering programs to be as flexible as possible since program goals or objectives vary widely from program to program. To implement this principle, a distinction is made among the KUs by identifying those that are *core* or essential units to the curriculum compared to those that are *supplementary* or extra units. Core components comprise knowledge and skills for which there is broad consensus that anyone obtaining a four-year degree in the field should acquire. Supplementary components comprise knowledge and skills that reflect expectations for advanced work according to the needs of a program.

TABLE III. CE2016 KNOWLEDGE AREAS

Tag	*Knowledge Area*	*Core Hours*
CE-CAE	Circuits and Electronics	50
CE-CAL	Computing Algorithms	30
CE-CAO	Computer Architecture and Organization	60
CE-DIG	Digital Design	50
CE-ESY	Embedded Systems	40
CE-NWK	Computer Networks	20
CE-PPP	Preparation for Professional Practice	20
CE-SEC	Information Security	20
CE-SGP	Signal Processing	30
CE-SPE	Systems and Project Engineering	35
CE-SRM	Systems Resource Management	20
CE-SWD	Software Design	45
	Total	**420**

To give readers a sense of the time required to cover a knowledge unit, CE2016 expresses time in hours, specifically in *core hours*. Core hours are suggested time of initial exposure to new material and do not assume any particular delivery method (traditional lecture, video lecture supporting a flipped classroom, directed study, seminar, etc.). As shown in Table II, the core hours of the BoK total 420, which is the equivalent of one academic year, leaving ample time for additional coverage of core and elective KUs, math and science, and general studies. Thus, a program may tailor the curriculum to meet its own unique objectives. In working with the CE2016 guidelines, the following points need emphasis.

- The core components refer to the knowledge and skills all students in all computer engineering degree programs *should* attain. Absence of some learning outcomes among the core components does not imply a negative judgment about their value, importance, or relevance. Rather, it simply means that the learning outcome is not expected of *every* student in *all* CE degree programs.
- The knowledge areas are *not* courses and the core components do *not* constitute a complete curriculum. Each program may choose to cover the core knowledge units in a variety of ways.
- Additional technical areas, as well as supporting mathematics, science and general studies, are necessary to produce a competent computer engineer.
- It is not the case that a program should achieve core knowledge units only within a set of introductory courses early in the four-year curriculum. While some core knowledge units are introductory, a program can address some core KUs only after students have developed significant background in their studies.

Beyond the computing knowledge areas, CE2016 recommends that a robust computer engineering program have at least four areas of capability that require at least 120 hours in mathematics to produce a competent computer engineering professional for the 2020s, although programs typically include much more mathematics to achieve their goals. The four areas, for which core knowledge units are defined in the report, include analysis of continuous functions (calculus), discrete structures, linear algebra, and probability and statistics. These emphasize what the steering committee considers essential to computer engineering.

IV. COMPLETING THE COMPUTER ENGINEERING CURRICULUM

By its very nature a computer engineering program should reflect an engineering ethos that permeates all years of the curriculum in a consistent manner. Such an approach has the effect of introducing students to engineering (and, in particular, computer engineering), teaching them to think and function as engineers, and setting expectations for the future. Preparation for practice is essential because most graduates from four-year programs directly begin professional practice in much of the world. Therefore, CE2016 goes beyond defining the body of knowledge to discuss integration into the curriculum of engineering design, laboratory experience, problem solving and critical thinking skills, as well as providing the background to adapt to new and emerging technologies in an agile manner. In addition, personal (soft) skills, oral and written communication skills, teamwork, lifelong learning skills, and professionalism are fundamental to preparation for professional practice of computer engineering.

Design must pervade the entire computer engineering curriculum, with students encountering different approaches to design so that they become familiar with the strengths and weaknesses of these approaches. The concept of a culminating design project is widely valued as an important experience occurring toward the end of a curriculum. Students consider a significant problem associated with a discipline and, in solving the problem, they demonstrate their ability to apply methodically engineering principles to generate a solution. For computer engineering, the solution typically involves the design and implementation of a system or subsystem containing both hardware and software components and considering a variety of interactions and tradeoffs.

Laboratory experiences are likewise an essential part of the curriculum. It is important that computer engineering students have many opportunities to observe, explore, and manipulate characteristics and behaviors of actual devices, systems, and processes. This includes designing, implementing, testing, and documenting hardware and software, designing experiments to acquire data, analyzing and interpreting that data, and using that data to correct or improve the design and to verify it meets specifications. To support such experiences CE2016 includes an appendix that describes possible laboratory configurations useful for developing modern student laboratory experiences for computer engineering programs.

Since computer engineering is a profession, CE2016 also addresses incorporation of professional issues throughout the curriculum. It is important that the curriculum include a wide range of professional practice issues, including consideration of the social context in which graduates implement engineering designs, legal and ethical issues and codes, and professionalism in the workplace.

V. Sample curriculum implementations

To illustrate how the guidelines of CE2016 might be applied within different institutional contexts, the report provides several sample curricula implementations for computer engineering programs. To provide a framework for a curriculum that illustrates the ideas presented in this report, the first three examples assume the following.

- Each year consists of two semesters with a student studying four to five modules (courses) per semester. Each module contains approximately 42 hours of instruction.
- Students should experience at least two computer engineering modules in the first year of study, at least four in the second year of study, and at least five in each of the third and fourth years of study.

Many institutions in the United States use the above pattern; the same is true in many other parts of the world. The fourth example of a curriculum implementation represents a typical four-year program in China. The remaining example reflects possible curricula reflecting those compatible with the Bologna Declaration in Europe [5].

All sample curricula in this appendix use a common format with five components.

1. A set of educational objectives for the program of study and an explanation of any assumed institutional, college, department, or resource constraints
2. A summary of degree requirements, in tabular form, to indicate the curricular content in its entirety
3. A sample schedule that a typical student might follow
4. A map showing coverage of the Computer Engineering Body of Knowledge by courses in the curriculum
5. A set of course descriptions for those courses in the computing component of the curriculum

Appendix A of this paper presents a sample four-year computer engineering curriculum model that might be administered by an electrical and computer engineering (ECE) department. The table in Appendix B indicates how the core knowledge units of the CE2016 BoK are covered in this curriculum. Each column of the table corresponds to one of the twelve BoK knowledge areas, and each row corresponds to a required course in the curriculum. The numbers in the table indicate specific knowledge units of that knowledge area that are covered in that course. For example, knowledge units 1-8 of the Embedded Systems knowledge area are covered in ECE_A202, with ECE_A403 covering knowledge units 9-13. A short description of each course is also provided with the sample curriculum. For example:

ECE_A202: Microprocessors

Architecture of microprocessor-based systems; study of microprocessor operation, assembly language, arithmetic operations, and interfacing.

Prerequisite: ECE_A201, CSC_A201
Credit Hours: 4 Lecture Hours: 42 Lab Hours: 42
BOK Coverage: CE-ESY 1-8, CE-CAO 3, 5, CE-DIG 3

VI. Conclusion

The CE2016 guidelines support the design of computer engineering curricula that will prepare graduates to function at entry-level positions in industry for continued career growth or to enter graduate programs for advanced study. The recommendations reflect input from industrial and educational institutions. This report is the result of a cooperative global effort of the professionals involved. Its intent is to provide interested parties and educational institutions worldwide a flexible way to implement a strong program in computer engineering. The CE2016 steering committee trusts that we have achieved that goal.

References

[1] J. Impagliazzo et al. "Curriculum guidelines for undergraduate degree programs in computer engineering," (CE2016). December 15, 2016. (http://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf)

[2] D. Soldan et al. "Curriculum guidelines for undergraduate degree programs in computer engineering," (CE2004). December 12, 2004. (http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf)

[3] ACM education website. (http://www.acm.org/education/curricula-recommendations)

[4] B.S. Bloom, ed., *Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain*, Longmans, 1956.

[5] European Commission/EACEA/Eurydice, 2015, *The European Higher Education Area in 2015: Bologna Process Implementation Report*, Luxembourg: Publications Office of the European Union..

Appendix A *Four-Year Model for Curriculum A*

CE: can be offered in computer engineering department
ECE: offered in the electrical and computer engineering department
CSC: offered in the computer science department
*CE technical electives: approved elective course in either department

Course	*Description*	*Credit*	*Course*	*Description*	*Credit*
	Semester 1			Semester 2	
MTH 101	Calculus I	3	MTH 102	Calculus II	3
CHM 101	Chemistry I & Lab	4	PHY 101	Physics I	3
CSC_A101	Introduction to Computer Programming	4	CSC_A102	Intermediate Computer Programming	4
	English Composition I	3	ECE_A101	Introduction to ECE	2
	Humanities Elective	3		English Composition II	3
	Total Credit Hours	17		*Total Credit Hours*	15
	Semester 3			Semester 4	
MTH 201	Calculus III	3	MTH 203	Differential Equations	3
PHY 201	Physics II	3	MTH_A204	Discrete Structures	3
ECE_A201	Digital Devices & Lab	4	ECE_A202	Microprocessors & Lab	4
CSC_A201	Data Structures	3	ECE_A203	Circuits/Electronics I	3
MTH 202	Linear Algebra	3	MTH 205	Probability & Statistics	3
	Total Credit Hours	16		*Total Credit Hours*	16
	Semester 5			Semester 6	
CSC_A301	Algorithms	3	CSC_A302	Client/Server Programming	3
ECE_A301	Circuits/Electronics II & Lab	4	ECE_A303	Signals & Systems	3
ECE_A302	Digital System Design & Lab	3	ECE_A304	Data Communication	3
	Humanities Elective	3	ECE_A305	Computer Architecture	3
				Social Science Elective	3
	Total Credit Hours	13		*Total Credit Hours*	15
	Semester 7			Semester 8	
ECE_A401	CE Design I	2	ECE_A402	CE Design II	2
ECE_A403	Embedded Systems & Lab	3	ECE_A404	Computer Security	3
ENG 401	Writing for Engineers	3	ECE_A405	Operating Systems	3
	CE Technical Elective*	3		CE Technical Elective*	3
	Fine Arts Elective	3		Social Science Elective	3
		14			14

Appendix B. *Mapping of Computer Engineering BOK to Curriculum A*

BOK Area / Course	CAE	CAL	CAO	DIG	ESY	NWK	PPP	SEC	SGP	SPE	SRM	SWD	ACF	DSC	LAL	PRS
Minimum Core BOK Hours	**50**	**30**	**60**	**50**	**40**	**20**	**20**	**20**	**30**	**35**	**20**	**45**	**30**	**30**	**20**	**30**
CSC$_A$101												1-4				
CSC$_A$102												4-8				
CSC$_A$201												5-9				
CSC$_A$301		1-8														
CSC$_A$302										1-12		10,12				
ECE$_A$101							1-3,5									
ECE$_A$201				1-9												
ECE$_A$202			3, 5	3	1-8											
ECE$_A$203	1-4															
ECE$_A$301	5-10															
ECE$_A$302				1, 2, 6-11												
ECE$_A$303									1-7							
ECE$_A$304						1-11										
ECE$_A$305			1-11													
ECE$_A$401							1-6, 11			7-10						
ECE$_A$402							7-11			10-12						
ECE$_A$403					9-13					10	4,6	11				
ECE$_A$404								1-11								
ECE$_A$405		9									1-8					
MTH 202															1-10	
MTH 203													1-7			
MTH$_A$204														1-9		
MTH 205																1-10
Core BOK Units Covered	**1-10**	**1-8**	**1-11**	**1-11**	**1-13**	**1-11**	**1-11**	**1-11**	**1-7**	**1-12**	**1-6**	**1-10**	**1-7**	**1-9**	**1-10**	**1-10**
Supplementary BOK Units		**9**									**7-8**	**11,12**				