IF Programming Concepts are eliminated from the Curriculum THEN Civil Engineers will …

Charles Newhouse, Tanjina Afrin, Andrei Ramniceanu, and Matthew Swenty
Virginia Military Institute

Abstract

Since the mid 1980's, many Civil Engineering (CE) programs have included computer programming concepts as full courses or as parts of introductory courses. Recently, many CE programs have reduced time spent on programming or dropped the concept entirely resulting in only about fifty-percent of departments currently requiring computer programming.[1] The Virginia Military Institute (VMI) has followed national trends and finds itself inside the proverbial diamond-shaped decision block debating the pros and cons of including programming concepts. The Fundamentals of Engineering Civil Exam still assesses Computational Tools, but a search for the word "programming" in the recent "Civil Engineering Body of Knowledge – Third Edition" produces no results. ABET's most recent student outcomes do not specifically mention computer programming. As humanities and social science courses are added to curricula, should programming be dropped? This paper describes VMI's decision process up to this juncture and provides arguments for keeping some programming concepts.

Keywords

Programming, Civil Engineering, Curriculum, Decision Logic.

Current State of Programming in the CE Curriculum

Civil Engineering undergraduate curricula are currently under pressure to add courses that prepare students for employment while adequately covering the fundamental engineering concepts needed to ensure that engineering designs are safe for the public. As employers and agencies such as the American Society of Civil Engineering (ASCE) and the Accreditation Board for Engineering and Technology, Inc. (ABET) have lobbied for more writing, humanities, and social science courses, programs have had to sacrifice some technical courses to make room for the additional breadth of courses.

Some civil engineering programs have replaced computer programming courses with courses in engineering or math that help the department become accredited through ABET. Other CE programs have eliminated computer programming courses because they believe that using spreadsheets or commercially produced software adequately prepares students for a career in civil engineering. There is no question that achieving accreditation and providing courses that teach students skills that an employer values are both important and essential. The question is whether or not the skills learned in computer programming courses are still needed in the twenty-first century. Also, programs are under constant pressure to reduce the number of credit hours required to graduate while meeting the needs of the students, employers, and accrediting bodies.

CE departments are at a crossroads. Recent studies[1] and a review of programs that are in VMI's region indicate that approximately half of the CE curricula include some type of computer programming concepts. In order to make an informed decision on whether or not programming should be in a CE curriculum, a general history of computer programming in CE curricula is briefly presented along with the history of computer programming at VMI.

Brief History of Computer Programming in CE Curricula

Many are surprised that the use of computers by civil engineers began as early as the 1950s. Two major fields pushed the introduction of computers in civil engineering: the defense and space programs and the development of the interstate highway system. The first computers with basic programming tools were not user friendly and were only accessible enough to help civil engineers with specific applications. To help overcome this problem, IBM introduced the first practical FORTRAN compiler for small computers around 1960, which eventually became popular among the civil engineering firms and colleges[2]. Similar to many established engineering departments, the VMI CE department adopted computer programming shortly after this tool was available, as shown in the photo from 1966 in Figure 1.

Figure 1 – IBM Model 1620 shown in 1966 (courtesy of Cadet Kershaw, shown in the photo)

As engineering firms often required in-house software development, they sought engineering graduates with knowledge of fundamental computer science. To keep pace with the demand, experts advised colleges to add new courses beyond the introductory programming language course, which would be taught by educators with extensive computer science background[3]. Some CE departments heeded the advice while others did not. A survey conducted in 1986 by the ASCE Education Committee of the Technical Council on the Computer Practices revealed that most of the civil engineering departments only had one required programming course. [4]

An additional goal of including computer programing concepts was teaching the logic of the problem-solution process.[3] The ability to solve problems has always, and continues to be, one of the primary differences between engineers and other disciplines. The use of computers to aid in the design of civil engineering structures was taken seriously, as detailed by the ASCE Computer Practices Committee of the Technical Council on Computer Practices in 1979:

> "The engineer and his firm accept the professional responsibility, liability, and risks associated with computer-aided design when they affix their seal to drawings and specifications. Therefore, to ensure reliability, the engineer must develop competence in the use of judging computer programs through proper education and training."[3]

Slowly, computing in civil engineering moved away from in-house software development towards acquired third party software[2]. To assess the needs of computing curriculum in civil engineering, the ASCE task committee on computing education conducted another survey in 1988 among educators and practitioners. Both groups ranked 1) spreadsheet, 2) computer aided design and drafting (CADD), 3) programming, and 4) graphics and databases as the most needed computing skills in CE curricula[5]. The relative importance of these computing skills has remain unchanged[6,7,8,9]. However, over the past decades, the importance and use of Geographic Information System (GIS) and discipline-specific engineering software have increased. Starting around the early 1990's, the importance of teaching programming has declined among educators and practitioners[8]. But, a recent survey in 2012 did show a slight increase in the importance of teaching programming skills for one area of CE, collaborative environment subjects[10].

There are a number of possible reasons for the decline in teaching programming in CE curricula. They include the extensive use and computing power of spreadsheets, the perceived lack of relevance of traditional programming instruction to engineering practice, the development of shorter undergraduate engineering degrees that have fewer credit to devote to topics such as programing, and the unavailability of educators who are fluent in traditional programing methods[5,10]. Many skills, such as the need to access databases still exist (called data mining today), but engineers have begun to delegate this work to non-engineers on staff.

Today, most educational groups that influence the civil engineering programs do not emphasize computer programing. A review of the recently published Civil Engineering Body of Knowledge does not require programing as an outcome. Items such as "logical reasoning", "numerical methods", and "computer and information science to support analysis" are mentioned under the mathematics and natural science outcomes. However, they are listed as skills that will be acquired as part of the mastery of other outcomes.[11] Another influence, ABET, has no direct mention of computer programing in the general criteria. The old general criteria that was sunset in 2018-2019 used to read in item "k" that students should obtain "an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice."[12] The updated general criteria from 2019-2020 reads that students should obtain "an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics."[13] This change further supports the transition from teaching pure programming to a computational tools class that focuses on spreadsheets or math software. Clearly, the educational influences on CE programs are not emphasizing traditional programing in 2020.

Another set of entities that influence civil engineering curriculum are state licensing boards. Every state and territory has a licensing board that is commissioned by their legislative body. These boards are tasked with enforcing a set of regulations based upon the laws in their jurisdiction. Each of these jurisdictions are different, but many have requirements for the program content in engineering degrees. For example, the state of Illinois includes in their state statute that a graduate with a Bachelor of Science degree who wants to become licensed should have taken courses in "engineering and computer science."[14] The state of Maryland defines an engineering curriculum for a current graduate to have "…at least one high-level computer language such as FORTRAN, PASCAL, C/C++, or MATLAB so that the student is able to compose computer programs to solve problems in science and design."[15] Graduates from programs that do not include computer programming classes violate some state laws and regulations.

There are many groups that help direct the evolution of civil engineering program curriculum. They are not all in agreement as to whether computer programing is essential for the next generation of graduates. Many directors of civil engineering programs are at a cross roads and are trying to determine if changes are warranted in order to include a programming course in their curriculums.

Brief History of Computer Programming in the CE department at VMI

Civil Engineering has been taught at VMI since as early as 1841, as evidenced by a text book written by Professor Thomas Williamson entitled "An Elementary Course of Architecture and Civil Engineering."[16] Since VMI Civil Engineering existed well before computer programming was invented, a review of VMI's history of teaching computer programming is instructive and useful to illustrate the typical changes that have occurred in many established CE programs.

The CE department at VMI began requiring computer programming courses in 1964. The first course, EG 102 – Introduction to Computer Programming, had the description "Fundamentals of Programming in FORTRAN language for the IBM 1620 Digital Computer as applied to Engineering Problems." From 1964 till the present, a stretch of 55 years, the CE department has maintained some computer programming requirement in the curriculum, as shown in Figure 2.

As shown in Figure 2, the CE department actually required three 1-credit computer programming courses from the mid 1980's to the early 1990's. Originally, the engineering departments offered the computer programming courses because the punch-card operated IBM 1620 computer was managed by the engineering departments. As the computer science (CS) department developed, many of these courses were taken over and offered as CS courses.

Around 1995, the CE department began to offer the programing courses, the first listed as CE 108 – Civil Engineering Computing. This course's description indicated that it covered "Advanced applications of electronic spreadsheets for engineering problem solving and presentation of results (and) programming techniques using a higher level programming language." This was the first mention of spreadsheets.

Prior to 1982, all students at VMI (called cadets) took a general introduction to engineering course. The course CE 101 Introduction to Civil Engineering was added in 1982 and included

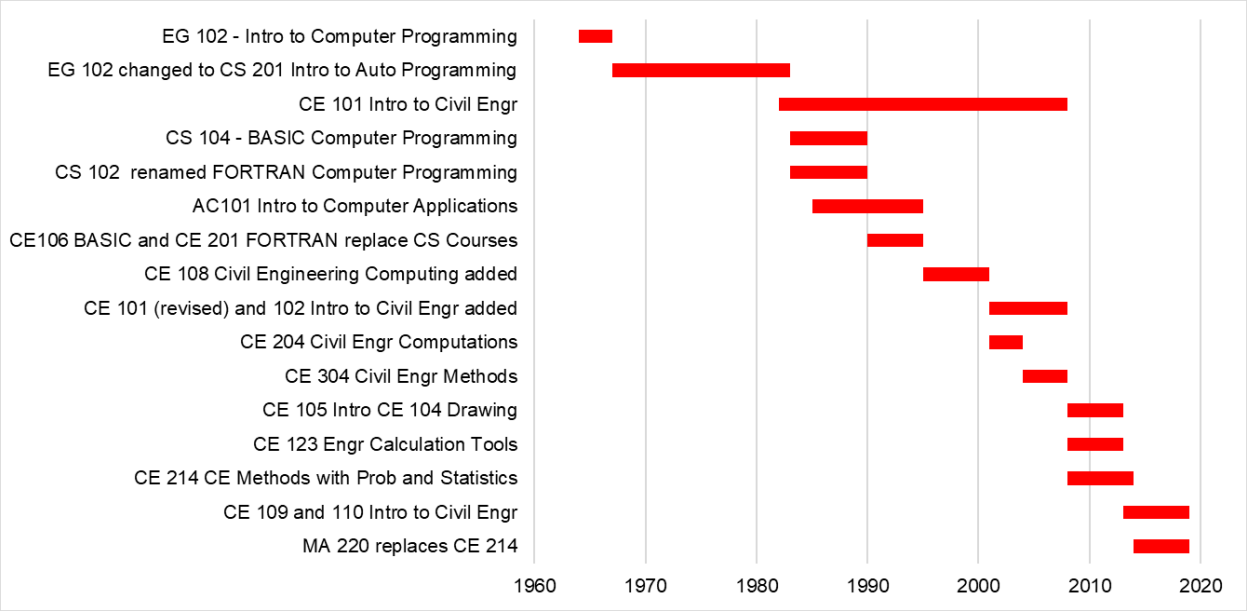

Figure 2 – Courses with Computer Programming Concepts at VMI through the Years

"A study of the engineering methods to include programming in the BASIC language." This was the first time that computer programming concepts were included in an introduction course. As shown, this course overlapped with CS 104 – BASIC Computer Programming. It is likely that the course description was not updated when CS 104 became a required course.

The required math courses are not included in Figure 2. But, for approximately a decade, students were also required to take Finite Math, which had a description of "Topics covered will be set theory, probability, systems of equations and matrices. Where appropriate, the cadet will construct and run computer programs. The computer applications will be chosen to enhance the cadet's understanding of the topics being studied." Without a doubt, from the mid 1960's to the early 2000's, cadets were required to take multiple courses that included computer programming concepts. Also, additional courses in the curriculum (including the Soils course) required cadets to write programs as class assignments.

In the mid 2000's, the inclusion of spreadsheets began to displace computer programming concepts. Courses with titles such as Civil Engineering Calculation Tools started to be required. These courses often included both spreadsheet concepts and statistics. More recently, the CE department at VMI has gone back to require a math course, MA 220, which is a more traditional probability and statistics course. Today, in 2019, the course descriptions for the two-two credit intro courses CE 109 and 110 include both spreadsheets and computer programming in either MATLAB® or Mathcad. However, not every instructor believes in including MATLAB® or Mathcad.

Current State of Computer Programing in CE Curricula

Recent surveys indicate that approximately half of the CE departments currently require some form of computer programming to complete degree requirements.[1] To confirm this regionally, the authors reviewed the 13 colleges/universities that along with VMI are members of the ASCE

Virginias' conference. This conference meets yearly to compete in various events including the national ASCE Concrete Canoe competition. All members of the conference offer either a technology or bachelor's degree in Civil Engineering. This cohort of schools includes small rural colleges, urban schools, and large flagship universities.

As shown in Table 1, six of the schools currently do not offer some form of computer programming while seven of the schools do offer computer programming as a civil engineering degree requirement. This information was gathered by reviewing the respective course catalogues published on the websites of the member schools in the fall of 2019. It is possible that the information on the website is slightly outdated; however, ABET requires proper distribution of the course descriptions so they are likely no more than a couple years old. The information obtained does confirm the previously reported national data obtained and presented at the 2018 national ASEE conference.[1]

No definitive conclusions can be drawn from Table 1 based on the size of the institution. There is a slight tendency for larger institutions to require computer programming, but there are exceptions such as West Virginia University. The technology programs, Bluefield State and Fairmont State, do not require programming. All of the universities in the Washington DC metro area offer a programming course. With a small sample size it is difficult to draw any conclusions from these patterns besides requiring a programing course is not universal.

Table 1 – Virginias' Conference Member Institutions and Programming Course Option

Institution	Course Offered	Institution	Course Offered
Bluefield State College	No	Catholic University of America	Yes
Fairmont State College	No	George Mason University	Yes
Marshall University	No	George Washington University	Yes
Old Dominion University	No	Howard University	Yes
West Virginia University	No	University of the District of Columbia	Yes
West Virginia University Institute of Technology	No	University of Virginia	Yes
		Virginia Tech	Yes

Based on national surveys and regional institutional offerings, it is clear that approximately half of the CE programs still require some form of computer programing.

Reasons for Requiring Computer Programming Concepts

Conversations with students and professors that experienced the initial inclusion of programming concepts into CE curriculum revealed that most believed that computers would be the way of the future. There was a common feeling that civil engineering students needed to be leaders in this new field. This was, and to many, still is a good argument. Overwhelmingly, reports from alumni confirm that students who graduated with some computer programming knowledge were valued by employers because they did not need to be specifically trained on the job to perform higher level computer application tasks.

In the early years, many civil engineering design firms wrote their own programs to aid in the design process. This process required engineers to interpret the design codes correctly and create

programs that would work correctly. Many regional and mid-size companies had robust in-house design computer programs that were created by and coded by civil engineering graduates. The employer demand for engineering graduates with programming skills was certainly prevalent in the early days.

As personal computers and network storage (and later cloud storage) became more common in engineering firms, many began to transition away from using in-house programs and started using commercially available software. For anyone who worked in a design firm during this transitional time, this was a two-edged sword. The early commercially available software was often capable of performing many tasks, but often could not be modified for specific design problems. Employers began to look for graduates that could use software instead of create software. More recently, as commercial software companies have been competing with each other, the race is on to make the software do everything. This has increased the size and complexity of the software to a point where it has become almost impossible to determine whether or not the software is working correctly without performing an independent check. This has produced a need for graduates to spend more of their time learning complex software and less time on fundamental engineering concepts and design codes. This shift in demand continues today.

Is it acceptable for a CE graduate to be a user of programs (commercial software) instead of a creator of programs? Many employers will argue that they need engineers that can hit the ground running and input designs into a software package. They say there is no need for an understanding of the programming concepts that were used to create the software. Others argue that the best, and at times only way, to fully understand a concept is to either teach it or to write a program that will perform the concept. Both of these methods have been proven to be excellent ways to teach complex concepts. Since not every CE graduate will go on to teach, the only and often best way to ensure that students can grasp complex concepts is to require the students to create programs that use these complex concepts.

Another benefit of learning some computer programming concepts is learning decision logic. Many engineers do not take logic courses because most curricula consider logic courses as math courses. Meeting the ABET requirements for math courses usually fills up the math course requirements in the curriculum with more traditional math courses (calculus, differential equations, linear algebra). Therefore, at best, a course in logic could only be counted as a free elective, or would not count at all. Many students use high school Advance Placement credits for their free electives, and therefore do not have a place to take a logic course. However, learning some computer programming does expose students to concepts of decision logic, which are valuable to employers.

Conclusion

Demand of programming courses in the CE curricula has declined over the last few decades due to several reasons such as perceived lack of relevance, shrinking space in the curricula, and transition to spreadsheets and commercial engineering software packages. This paper reviewed VMI's cohort schools' CE curricula and found approximately half of the schools do not require some type of computer programming concept course, confirming national trends. Exploring VMI's programming course history revealed that for a long time, students were required to take

multiple courses that included computer programming concepts. However, after the mid-2000's, spreadsheet concepts, computer aided drafting, and statistics slowly began to replace the programming concept courses following the national trend. This paper provides arguments for the necessity of the skill set learned in computer programming courses in CE curricula.

Programming concepts are needed in the rapidly advancing engineering world and help provide a more thorough understanding of the fundamental engineering concepts students need upon graduation. If civil engineering programs continue to deemphasize computer programing, then there will be a generation of graduates who may not fully understand the nature and risks involved with using complex commercial computer software. One of the best methods of fully understanding complex engineering concepts is to study the concept deep enough to be able to write a program to solve the problem. The appearance that students are better prepared for the workforce by understanding how to use complex commercial design software instead of understanding how programs work is a misguided short term tradeoff. Graduates that thoroughly understand the engineering concepts and decision logic that form the basis of engineering will be better prepared to safely solve the problems of tomorrow.

References

1. Swenty, Brian and Swenty, Matthew, "The Impact of the EAC-ABET Program Criteria on Civil Engineering Curricula," ASEE Annual Conference and Exposition, Salt Lake City, 2018.
2. `Fenves, S. J., & Rasdorf, W. J., "Role of ASCE in the Advancement of Computing in Civil Engineering." Journal of computing in civil engineering, 15(4), ASCE, 2001, 239-247.
3. Rasdorf, W. J., "Computer programming in the civil engineering curriculum." Journal of Professional Issues in Engineering, 111(4), ASCE, 1985, 141-148.
4. "Computing training of civil engineers". Panel Session, National Spring Convention, ASCE, Atlantic City, N.J., 1987.
5. Law, K. H., Rasdorf, W. J., Karamouz, M., & Abudayyeh, O. Y., "Computing in civil engineering curriculum: Needs and issues." Journal of professional issues in engineering, 116(2), ASCE, 1990, 128-141.
6. O'Neill, R. J., Henry, R. M., & Lenox, T. A. (1996a). "Role of Computing: Practitioners' Perspective." Computing in Civil Engineering, ASCE, 1996a, 670-676.
7. O'Neill, R. J., Henry, R. M., & Lenox, T. A. (1996b). "Role of Computing: Educators' Perspective." Proceedings of the ASEE Annual Conference, 1996b.
8. Abudayyeh, O., Cai, H., Fenves, S. J., Law, K., O'Neill, R., & Rasdorf, W., "Assessment of the computing component of civil engineering education." Journal of computing in civil engineering, 18(3), ASCE, 2004, 187-195.
9. Gerber, D. J., Khashe, S., & Smith, I. F., "Surveying the evolution of computing in architecture, engineering, and construction education." Journal of Computing in Civil Engineering, 29(5), ASCE, 2013.
10. Khashe, S., Gerber, D. J., & Smith, I. F., "Surveying the evolution of computing in architecture, engineering, and construction education since 2012." Journal of Computing in Civil Engineering, 30(6), ASCE, 2016.
11. Civil Engineering Body of Knowledge 3 Task Committee, "Civil Engineering Body of Knowledge." American Society of Civil Engineers, 3, ASCE, 2019.
12. Accreditation Board for Engineering and Technology, "Criteria for Accrediting Engineering Programs, 2018-2019." Accessed 11/6/19. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2018-2019/
13. Accreditation Board for Engineering and Technology, "Criteria for Accrediting Engineering Programs, 2019-2020." Accessed 11/6/19. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/#GC3
14. State of Illinois Statutes., "Chapter 225. 325. Section 4. Definitions (r)." Accessed 1/8/20. http://www.ilga.gov/legislation/ilcs/ilcs.asp

15. State of Maryland Procedural Regulations. “09.23.05.02 Engineering Curriculum.” Accessed 1/8/20. http://www.dsd.state.md.us/comar/SubtitleSearch.aspx?search=09.23.*
16. Textbook, "An Elementary Course of Architecture and Civil Engineering.” Thomas H. Williamson Papers, Virginia Military Institute Archives. https://archivesspace.vmi.edu/repositories/3/archival_objects/2064 Accessed January 01, 2020

Charles Newhouse

Charles earned his Ph.D. in Civil Engineering at Virginia Tech after working nine years as a consulting structural engineer for MMM Design Group in Norfolk, Virginia. He spent three years teaching at Texas Tech University before joining the faculty at VMI in 2008 where he is the Charles S. Luck, Jr. ’20 Institute Professor in Engineering. He also currently serves as a Special Assistant to the Dean for Permits and Calendar issues.

Tanjina Afrin

Tanjina Afrin earned her PhD from Clemson University. She is currently working as an Assistant Professor at Virginia Military Institute (VMI). She teaches basic fluid mechanics, water resources engineering, and hydrology. She enjoys helping students with their class/research projects.

Andrei Ramniceanu

Andrei Ramniceanu earned his BS, MS and PhD in civil engineering from Virginia Tech. Following graduation Andrei worked at Professional Service Industries, Inc. as a petrographer. He is currently an assistant professor of Civil Engineering at the Virginia Military Institute where he teaches materials related courses. He enjoys working with the students on concrete related projects and Concrete Canoe.

Matthew Swenty

Matthew (Matt) Swenty obtained his Bachelors and Masters degrees in Civil Engineering from Missouri S&T and then worked as a bridge designer at the Missouri Department of Transportation. He went on to obtain his Ph.D. in Civil Engineering at Virginia Tech and worked at the Turner-Fairbank Highway Research Center on bridge research projects. He is currently an associate professor of Civil Engineering at the Virginia Military Institute (VMI). He teaches engineering mechanics and structural engineering courses and enjoys working with the students on bridge related research projects and with the ASCE student chapter.