2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #53028

Pre-College Experience Factors Contributing to Categories of Computational Thinking in an Introductory Engineering Course

Anne Hong, University of Cincinnati

Anne Hong is an engineering education PhD student in the Department of Engineering and Computing Education at the University of Cincinnati College of Engineering and Applied Science. She holds a BS in biomedical engineering and an MS in health informatics with a focus on data analysis and biostatistics from Rutgers University. Anne has been an instructor at the college level for over 6 years, teaching and facilitating a variety of courses for undergraduate students including physics, biology, and engineering economics. Her research interests are focused on examining how teaching pedagogy and curriculum designs in undergraduate introductory and pre-requisite engineering courses impact performance in upper-level engineering coursework.

Dr. So Yoon Yoon, University of Cincinnati

Dr. So Yoon Yoon is an assistant professor in the Department of Engineering and Computing Education in the College of Engineering and Applied Science at the University of Cincinnati, OH, USA. Dr. Yoon received her Ph.D. in Gifted Education and an M.S.Ed. in Research Methods and Measurement with a specialization in Educational Psychology, both from Purdue University, IN, USA. She also holds an M.S. in Astronomy and Astrophysics and a B.S. in Astronomy and Meteorology from Kyungpook National University, South Korea. Her work centers on elementary, secondary, and postsecondary engineering education research as a psychometrician, data analyst, and program evaluator with research interests in spatial ability, STEAM education, workplace climate, and research synthesis with a particular focus on meta-analysis. She has developed, validated, revised, and copyrighted several instruments beneficial for STEM education research and practice. Dr. Yoon has authored more than 100 peer-reviewed journal articles and conference proceedings and served as a journal reviewer in engineering education, STEM education, and educational psychology. She has also served as a PI, co-PI, advisory board member, or external evaluator on several NSF- and NASA-funded projects.

Dr. Noemi V Mendoza Diaz, Texas A&M University

Dr. Mendoza is a faculty member of Technology Management in the College of Education-Engineering at Texas A&M University. She has worked as electrical engineering professor in Mexico. She recently obtained funds from NSF to investigate enculturation to engineering and computational thinking in engineering students. She is the co-advisor of the Society for Hispanic Professional Engineers at TAMU and is interested in computing engineering education and Latinx engineering entrepreneurship.

Pre-College Experience Factors Contributing to Categories of Computational Thinking in an Introductory Engineering Course

Abstract

Computational thinking is a foundational skill in engineering education, closely tied to student success and widely taught in early coursework through programming instruction. This study examined the pre-college experience factors contributing to computational thinking skills of students enrolled in an introductory engineering course at a public research university in the Midwestern U.S. At the end of the Spring 2025 semester, a survey containing the Engineering Computational Thinking Diagnostic and questions about students' pre-college experiences was administered to students enrolled in the second of two first-year introductory engineering courses. Pre-college experiences examined included higher level math courses, computer science courses, engineering or computer science related career and technical education courses, related extracurricular activities, and the number of college credits earned before taking the course. Using responses from 785 participants, multiple regression analysis identified higher-level math coursework completed prior to college and the number of college credits earned before taking the introductory engineering course as significant predictors of computational thinking. In addition, this study revealed how significant pre-college experiences varied across the categories of the computational thinking framework. The results of this study inform educators in designing introductory engineering courses to develop computational thinking more effectively.

Introduction

Computational thinking is a crucial component of engineering education. It applies not only to computer science, but also to all disciplines within engineering. As computational tools are increasingly used in engineering to solve complex problems, computational thinking has become an essential component of the curriculum. In addition, due to its alignment with multiple student outcomes listed by the Accreditation Board for Engineering and Technology (ABET), it plays a significant role within undergraduate engineering curricula across the United States [1]. Therefore, the importance of teaching computational thinking in engineering is broadly recognized.

A strong correlation has been found between the computational thinking skills of first-year engineering students and their future performance in engineering programs [2]. Thus, developing computational thinking skills early in the curriculum may be a key objective of many introductory engineering courses. These introductory engineering courses, often among the first major-specific courses taken by first-year engineering students, typically teach basic programming skills as a means of developing computational thinking.

Prior studies have shown a positive association between computational thinking and factors such as previous programming experience [3]. However, few studies have specifically focused on examining a variety of pre-college experience factors among undergraduate first-year engineering students. Due to the recognized importance of developing computational thinking early in engineering curricula, a closer examination of this group of students would be beneficial.

Purpose of the Study

This study aims to examine the pre-college experience factors contributing to computational thinking skills in an introductory engineering course. Guiding research questions are: (1) How do pre-college experiences impact computational thinking skills? and (2) How do pre-college experiences influence each of the five computational thinking categories?

Literature Review

Although the specific definition of computational thinking varies across the literature, computational thinking has been broadly defined as a problem-solving approach that involves formulating problems and expressing their solutions in a way that can be carried out through information processing [4]. The importance of developing computational thinking has been examined closely in the literature, and researchers have proposed and explored different ways to develop and assess computational thinking. Multiple studies emphasized how computational thinking development should start at the K-12 level. Espinal et al. [5] explored the pedagogical approaches used by teachers to integrate computational thinking in K-12 classrooms. Tissenbaum et al. [6] suggested teaching computational thinking to K-12 students through computational action, where students learned computing through projects that were directly relevant to their own lives. Approaches for teaching computational thinking in higher education include modeling and simulation practices [7], [8], [9] and robotics tasks [10].

Several frameworks have been developed to characterize computational thinking; one of these frameworks was created by the International Society for Technology in Education (ISTE) to help educators incorporate computational thinking in the K-12 curricula [11]. Another computational thinking framework was created by the Collaborative Process to Align Computing Education with Engineering Workforce Needs (CPACE) to inform engineering curricula in higher education [12]. However, the scope of the CPACE framework is narrow, focusing on software and computing rather than computational thinking as a whole. Other computational thinking frameworks have been developed by the College Board for the AP Computer Science Principles (CSP) exam [13], the Computer Science Teachers Association for K-12 computer science standards [14], and Brennan & Resnick [15]. Building on established computational thinking frameworks in the literature, the Engineering Computational Thinking Diagnostic (ECTD) uses a specific framework tailored to computational thinking in engineering education, consisting of five categories: Abstraction; Algorithmic Thinking and Programming; Data Representation, Organization, and Analysis; Decomposition; and Impact of Computing. The alignment of the computational thinking framework used by the ECTD with other computational thinking frameworks is shown in Table 1. Figure 1 provides the definitions of the five categories [16].

Table 1. Engineering Computational Thinking Diagnostic (ECTD) framework compared with the literature (after Mendoza Diaz et al., 2023 [6])

ECTD	ABET	ISTE	CSP	CPACE	CSTA	Brennan & Resnick

<table>
<tr><td>Abstraction</td><td>1, 7</td><td>Abstraction</td><td>Abstraction</td><td>Modeling and Abstraction</td><td></td><td></td></tr>
<tr><td rowspan="4">Algorithmic Thinking and Programming</td><td rowspan="2">6, 7</td><td></td><td>Algorithms</td><td rowspan="2">Algorithmic Thinking and Programming</td><td rowspan="2">Algorithms and Programming</td><td></td></tr>
<tr><td></td><td>Programming</td><td>Multiple Programming Constructs</td></tr>
<tr><td rowspan="2">1, 7</td><td>Pattern Matching</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Automation</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">Data Representation, Organization, and Analysis</td><td rowspan="2">6, 7</td><td rowspan="2"></td><td rowspan="2">Data and Information</td><td>Digital Representation of Information</td><td rowspan="2">Data and Analysis</td><td>Data</td></tr>
<tr><td>Information Organization</td><td></td></tr>
<tr><td>Decomposition</td><td>1, 7</td><td>Decomposition</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">Impact of Computing</td><td rowspan="2">2</td><td rowspan="2"></td><td>Global Impact</td><td></td><td>Impacts of Computing</td><td rowspan="2"></td></tr>
<tr><td>The Internet</td><td>Networks</td><td>Networks & the Internet</td></tr>
</table>

Note. ECTD = Engineering Computational Thinking Diagnostic; ABET = Accreditation Board for Engineering and Technology; ISTE = The International Society for Technology in Education; CSP = The AP Computer Science Principles; CPACE = Collaborative Process to Align Computing Education with Engineering Workforce Needs; CSTA = Computer Science Teachers Association.

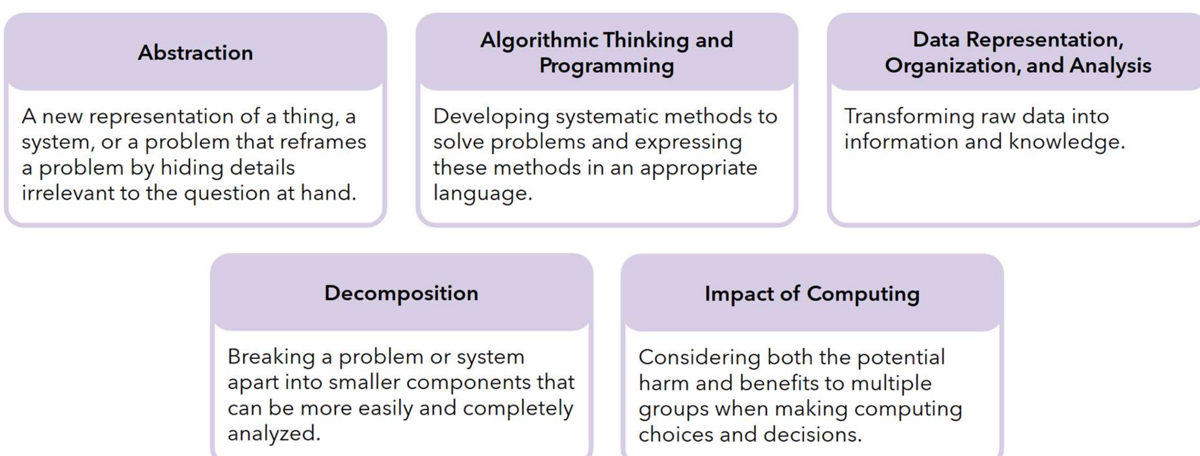


Figure 1. Definitions of the five categories of computational thinking.

Despite several studies using the ECTD framework to examine computational thinking, the extent to which each of these categories is developed in introductory engineering courses, as well as the varying factors that influence each category, remains unexplored [3], [17], [18].

Method

Setting

The study took place in the second in a sequence of two first-year introductory engineering courses at a large, urban, public research university in the Midwestern United States. These courses focus on developing skills in engineering problem solving, design, and computation, as well as communication, teamwork, and other professional skills.

The first course introduces students to the different engineering disciplines and the engineering design process. Topics covered in the course include spatial visualization, flow-charting, Excel, and Python. Throughout the course, in teams of approximately four, students participate in a semester-long project while learning to apply basic engineering principles to identify problems and construct solutions. This project is open ended and student-defined, focusing on exploring real world issues. The exams are comprehensive and cover the material taught throughout the course.

The second course builds on the foundational concepts taught in the first course of the two-course sequence. Students are introduced to various computational tools used in engineering problem solving and design, including LabVIEW, MATLAB, and Visual Basic, while learning concepts in statistics, statics, electric circuits, and material energy balance. Throughout the course, in teams of approximately four, students participate in a semester-long group project using Lego robotics. Like the first course in the sequence, this course also has two comprehensive exams, which are completed individually and in person.

Participants

The population in this study consists of students enrolled in the second of two first-year introductory engineering courses at a large, urban, public R1 university in the Midwestern United States. In Spring 2025, there were 17 sections of the course and a total of 8 instructors. The class size was capped at 72 students. A total of 1,193 students were originally enrolled in the course, of which 1,171 remained enrolled by the end of the semester. After IRB approval, participants were recruited through emails distributed to all students enrolled in the course at the end of the Spring 2025 semester. From the 1,001 total survey responses collected, valid responses from 785 students (67.0% response rate) were included in the study after duplicate responses, incomplete responses, and other invalid responses were removed. The demographic characteristics of these participants are shown in Table 2.

Table 2. Demographic Characteristics of Participants (*N* = 785)

Characteristic	*n*	%

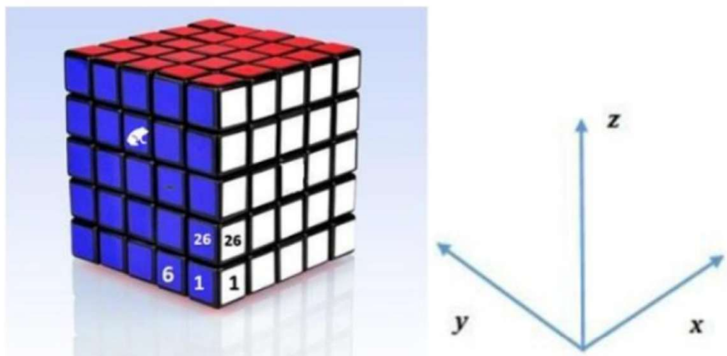
Gender		
Female	177	22.5
Male	608	77.5
Residency		
Domestic	709	90.3
International	76	9.7
Race/Ethnicity*		
American Indian or Alaska Native	0	0.0
Asian	51	7.2
Black or African American	27	3.8
Hispanic/Latino	46	6.5
Native Hawaiian or Other Pacific Islander	1	0.1
Two or More Races	36	5.1
White	542	76.4
Unknown	6	0.8
First-Generation Status		
Yes	101	12.9
No	665	84.7
Unknown	19	2.4
Total	785	100.0

Note. *Race/Ethnicity was categorized for domestic students only.

Measures

Computational thinking was measured using the Engineering Computational Thinking Diagnostic (ECTD) [16]. The instrument consists of 20 multiple-choice questions, with four questions about each of the five categories of computational thinking. Each question has six answer choices: one correct option, four incorrect options, and one “I don’t know” option. The ECTD is scored out of 20. In addition, each of the five categories is scored out of 4. The “I don’t know” option was scored as an incorrect response. The ECTD was administered online as part of a Qualtrics survey and was not timed. An example item from the ECTD is shown in Figure 2.

11. The Rubik's cube in the figure is composed of 125 blocks. A program counts the number of blocks from the origin of the coordinate system to a given block by counting the blocks along the x-axis, then along the y-axis, and finally along the z-axis. For example, the block labeled 6 is the sixth block counted and the block labeled 26 is the twenty-sixth block counted. How many blocks have been counted by the program when it arrives at the frog icon?



- 6
- 11
- 18
- 78
- 86
- I don't know

Figure 2. Example item from the ECTD. This item focuses on the Decomposition category of computational thinking.

Questions about pre-college experiences were included in the Qualtrics survey containing the ECTD. The questions about pre-college experiences asked about the following factors: (a) higher level math courses taken prior to college, (b) computer science courses taken prior to college, (c) engineering or computer science related career and technical education courses taken prior to college, and (d) math, computer science, or engineering extracurricular activities participated in prior to college. Higher level math courses included calculus, multivariate calculus, differential equations, linear algebra, and statistics. Algebra, geometry, trigonometry, and pre-calculus were not categorized as higher level math courses. Computer science courses included AP Computer Science Principles, Introduction to Programming, and other similar courses with a programming focus. Extracurricular activities included engineering clubs, robotics competitions, math teams, and other similar activities. An additional factor, the number of college credits completed prior to taking the course, was collected from the university archive.

Procedure

The Qualtrics survey containing the ECTD and questions about students' pre-college experiences was administered online to students near the end of the Spring 2025 semester. Information about student demographics and the number of college credits completed prior to taking the course was collected from the university archive.

Data Analysis

Standard statistical analysis was performed using Excel and *R*, such as descriptive statistics, correlations, disaggregated by pre-college experiences. Multiple regression analysis was conducted with ECTD score as the dependent variable and pre-college experiences as the independent variables. Further analysis was conducted with each separate category of the computational thinking framework as the dependent variable.

Results

Descriptive Statistics Results of ECTD Scores by Pre-College Experiences

A summary of ECTD scores disaggregated by pre-college experiences is shown in Table 3.

Table 3. Descriptive Statistics of ECTD Scores by Pre-College Experiences

Pre-College Experiences	Category	*n*	%	*M*	*SD*
Higher level math courses	Yes	571	72.7	12.85	4.81
	No	210	26.8	11.05	4.63
CS courses	Yes	206	26.2	12.76	5.18
	No	571	72.7	12.21	4.69
Engineering or CS related CTE courses	Yes	271	34.5	12.11	5.01
	No	494	62.9	12.48	4.73
Math, CS, or engineering extracurricular activities	Yes	337	42.9	12.72	4.94
	No	439	55.9	12.08	4.73
No. of college credits completed prior to taking the course	0 - 23	201	25.6	10.57	4.92
	24 - 47	340	43.3	12.41	4.55
	48 - 71	173	22.0	13.72	4.74
	72 - 92	54	6.9	14.24	3.55
	93+	17	2.2	12.41	6.76
Total		785		12.36	4.83

Note. CS = Computer Science; CTE = Career and Technical Education

A summary of ECTD category scores disaggregated by pre-college experiences is shown in Table 4.

Table 4. Descriptive Statistics of ECTD Category Scores by Pre-College Experiences

Pre-College Experiences	Category	*n*	%	ABS		ATP		DROA		DEC		IOC	
				M	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Higher level math courses	Yes	571	72.7	2.49	1.19	2.60	1.17	2.54	1.24	2.32	1.30	2.89	1.29
	No	210	26.8	2.06	1.20	2.29	1.17	2.09	1.21	1.90	1.28	2.71	1.33
CS courses	Yes	206	26.2	2.48	1.22	2.66	1.17	2.50	1.29	2.23	1.42	2.90	1.28
	No	571	72.7	2.34	1.20	2.47	1.18	2.39	1.23	2.19	1.27	2.82	1.31
Engineering or CS related CTE courses	Yes	271	34.5	2.30	1.24	2.46	1.19	2.30	1.25	2.18	1.32	2.86	1.28
	No	494	62.9	2.41	1.19	2.54	1.17	2.48	1.24	2.21	1.31	2.83	1.31
Math, CS, or engineering extracurricular activities	Yes	337	42.9	2.48	1.21	2.62	1.16	2.52	1.22	2.25	1.30	2.84	1.30
	No	439	55.9	2.29	1.19	2.44	1.19	2.34	1.26	2.17	1.32	2.84	1.30
No. of college credits completed prior to taking the course	0 - 23	201	25.6	1.96	1.16	2.19	1.18	2.00	1.23	1.93	1.40	2.49	1.44
	24 - 47	340	43.3	2.40	1.17	2.49	1.16	2.37	1.21	2.20	1.22	2.95	1.22
	48 - 71	173	22.0	2.69	1.18	2.84	1.15	2.77	1.20	2.43	1.32	2.98	1.24
	72 - 92	54	6.9	2.61	1.19	2.87	0.97	3.06	0.90	2.56	1.22	3.15	1.07
	93+	17	2.2	2.65	1.37	2.53	1.37	2.65	1.62	2.06	1.56	2.53	1.59
Total		785		2.37	1.20	2.52	1.18	2.42	1.25	2.20	1.31	2.84	1.30

Note. CS = Computer Science; CTE = Career and Technical Education; ABS = Abstraction; ATP = Analytical Thinking and Programming; DROA = Data Representation, Organization, and Analysis; DEC = Decomposition; IOC = Impact of Computing

Impact of Pre-college Experiences on Computational Thinking Skills

Table 5 shows the correlations between pre-college experiences and scores from the ECTD and all five ECTD categories. When examining the correlations between each variable independently, only higher level math courses taken prior to college and the number of college credits completed prior to taking the course are significantly correlated with ECTD scores.

Table 5. Correlations of Pre-College Experiences with ECTD Category Scores and Overall ECTD Scores

	ABS	ATP	DROA	DEC	IOC	ECTD
Higher level math courses (0 = no, 1 = yes)	**0.175***	**0.130***	**0.170***	**0.145***	0.062	**0.175***
CS courses (0 = no, 1 = yes)	0.054	**0.072***	0.035	0.013	0.025	0.050
Engineering or CS related CTE courses (0 = no, 1 = yes)	-0.038	-0.033	-0.069	-0.015	0.006	-0.038

Math, CS, or engineering extracurricular activities (0 = no, 1 = yes)	**0.087***	**0.081***	**0.076***	0.028	0.003	0.070
No. of college credits completed prior to taking the course (1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+)	**0.197***	**0.180***	**0.240***	**0.128***	**0.114***	**0.221***
ECTD	**0.695***	**0.778***	**0.824***	**0.796***	**0.773***	1.000

Note. **p* < 0.05; CS = Computer Science; CTE = Career and Technical Education; ABS = Abstraction; ATP = Analytical Thinking and Programming; DROA = Data Representation, Organization, and Analysis; DEC = Decomposition; IOC = Impact of Computing

As shown in Table 5, when examining the correlations between each variable independently, having taken higher level math courses before college is significantly correlated with all categories in the ECTD except for Impact of Computing (IOC). The number of college credits completed prior to taking the course is significantly correlated with all ECTD category scores. Having taken CS courses before college is significantly correlated only to the Algorithmic Thinking and Programming (ATP) category, and having participated in math, CS, or engineering extracurricular activities is significantly correlated to a few of the categories, though all these correlations are very weak.

Table 6 shows the multiple regression coefficients of pre-college experiences on ECTD scores. When pre-college experiences were considered to predict ECTD scores in multiple regression modeling, only higher level math courses and the number of college credits completed prior to taking the course were significant factors.

Table 6. Regression Coefficients of Pre-College Experiences on ECTD Scores (*n* = 757)

Variable	Unstandardized		Standardized	*t*	*p*	Collinearity	
	B	S.E.	*β*			Tolerance	VIF
(Constant)	8.97	0.49		18.22	< .001		
Higher level math courses	1.56	0.39	**0.144**	4.02	**< .001**	0.95	1.05
CS courses	0.18	0.40	0.016	0.45	0.654	0.95	1.05
Engineering or CS related CTE courses	-0.39	0.36	-0.038	-1.07	0.283	0.96	1.04
Math, CS, or engineering extracurricular activities	0.31	0.36	0.032	0.87	0.387	0.92	1.09
No. of college credits completed prior to taking the course	1.03	0.18	**0.204**	5.71	**< .001**	0.96	1.04

Note. R^2 = 0.077, Adjusted R^2 = 0.071; Higher level math courses: 0 = no, 1 = yes; CS courses: 0 = no, 1 = yes; Engineering or CS related CTE courses: 0 = no, 1 = yes; Math, CS, or engineering extracurricular activities: 0 = no, 1 = yes; No. of college credits completed prior to taking the course: 1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+; CS = Computer Science; CTE = Career and Technical Education

Impact of Pre-college Experiences on the Five Computational Thinking Categories

Tables 7-11 show the multiple regression coefficients of pre-college experiences on each of the five ECTD category scores. When pre-college experiences were considered to predict scores in the Abstraction category of the ECTD in multiple regression modeling, higher level math courses and the number of college credits completed prior to taking the course were significant factors, as shown in Table 7.

Table 7. Regression Coefficients of Pre-College Experiences on Abstraction of the ECTD (*n* = 757)

Variable	Unstandardized		Standardized	t	p	Collinearity	
	B	S.E.	β			Tolerance	VIF
(Constant)	1.59	0.12		12.83	< .001		
Higher level math courses	0.40	0.10	**0.147**	4.06	**< .001**	0.95	1.05
CS courses	0.02	0.10	0.006	0.17	0.863	0.95	1.05
Engineering or CS related CTE courses	-0.12	0.09	-0.049	-1.37	0.172	0.96	1.04
Math, CS, or engineering extracurricular activities	0.12	0.09	0.050	1.36	0.176	0.92	1.09
No. of college credits completed prior to taking the course	0.22	0.05	**0.175**	4.88	**< .001**	0.96	1.04

Note. R^2 = 0.068, Adjusted R^2 = 0.062; Higher level math courses: 0 = no, 1 = yes; CS courses: 0 = no, 1 = yes; Engineering or CS related CTE courses: 0 = no, 1 = yes; Math, CS, or engineering extracurricular activities: 0 = no, 1 = yes; No. of college credits completed prior to taking the course: 1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+; CS = Computer Science; CTE = Career and Technical Education

When pre-college experiences were considered to predict scores in the Algorithmic Thinking and Programming category of the ECTD in multiple regression modeling, as shown in Table 8, the higher level math courses and the number of college credits completed prior to taking the course were significant factors again.

Table 8. Regression Coefficients of Pre-College Experiences on Algorithmic Thinking and Programming of the ECTD (*n* = 757)

Variable	Unstandardized		Standardized	t	p	Collinearity	
	B	S.E.	β			Tolerance	VIF
(Constant)	1.84	0.12		15.12	< .001		

Higher level math courses	0.25	0.10	**0.095**	2.62	**0.009**	0.95	1.05
CS courses	0.12	0.10	0.046	1.26	0.208	0.95	1.05
Engineering or CS related CTE courses	-0.09	0.09	-0.035	-0.96	0.336	0.96	1.04
Math, CS, or engineering extracurricular activities	0.11	0.09	0.046	1.23	0.218	0.92	1.09
No. of college credits completed prior to taking the course	0.21	0.04	**0.170**	4.71	**< .001**	0.96	1.04

Note. R^2 = 0.053, Adjusted R^2 = 0.047; Higher level math courses: 0 = no, 1 = yes; CS courses: 0 = no, 1 = yes; Engineering or CS related CTE courses: 0 = no, 1 = yes; Math, CS, or engineering extracurricular activities: 0 = no, 1 = yes; No. of college credits completed prior to taking the course: 1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+; CS = Computer Science; CTE = Career and Technical Education

When pre-college experiences were considered to predict scores in the Data Representation, Organization, and Analysis category of the ECTD in multiple regression modeling, higher level math courses, engineering or CS related CTE courses, and the number of college credits completed prior to taking the course were significant factors, as shown in Table 9. However, unlike the higher level math courses and the number of college credits completed prior to taking the courses, which were positive predictors, engineering or CS related CTE courses were a negative predictor.

Table 9. Regression Coefficients of Pre-College Experiences on Data Representation, Organization, and Analysis of the ECTD (*n* = 757)

Variable	Unstandardized		Standardized	*t*	*p*	Collinearity	
	B	S.E.	*β*			Tolerance	VIF
(Constant)	1.56	0.13		12.31	< .001		
Higher level math courses	0.38	0.10	**0.138**	3.84	**< .001**	0.95	1.05
CS courses	-0.01	0.10	-0.002	-0.05	0.959	0.95	1.05
Engineering or CS related CTE courses	-0.18	0.09	-0.071	-1.99	0.047	0.96	1.04
Math, CS, or engineering extracurricular activities	0.12	0.09	0.047	1.29	0.196	0.92	1.09
No. of college credits completed prior to taking the course	0.28	0.05	**0.215**	6.04	**< .001**	0.96	1.04

Note. R^2 = 0.084, Adjusted R^2 = 0.078; Higher level math courses: 0 = no, 1 = yes; CS courses: 0 = no, 1 = yes; Engineering or CS related CTE courses: 0 = no, 1 = yes; Math, CS, or engineering extracurricular activities: 0 = no, 1 = yes; No. of college credits completed prior to taking the course: 1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+; CS = Computer Science; CTE = Career and Technical Education

As shown in Table 10, when pre-college experiences were considered to predict scores in the Decomposition category of the ECTD in multiple regression modeling, higher level math courses and the number of college credits completed prior to taking the course were significant factors.

Table 10. Regression Coefficients of Pre-College Experiences on Decomposition of the ECTD (*n* = 757)

Variable	Unstandardized		Standardized	t	p	Collinearity	
	B	S.E.	β			Tolerance	VIF
(Constant)	1.60	0.14		11.72	< .001		
Higher level math courses	0.39	0.11	**0.134**	3.64	**< .001**	0.95	1.05
CS courses	-0.02	0.11	-0.006	-0.16	0.872	0.95	1.05
Engineering or CS related CTE courses	-0.03	0.10	-0.009	-0.26	0.798	0.96	1.04
Math, CS, or engineering extracurricular activities	0.00	0.10	-0.002	-0.05	0.964	0.92	1.09
No. of college credits completed prior to taking the course	0.16	0.05	**0.114**	3.11	**0.002**	0.96	1.04

Note. R^2 = 0.035, Adjusted R^2 = 0.029; Higher level math courses: 0 = no, 1 = yes; CS courses: 0 = no, 1 = yes; Engineering or CS related CTE courses: 0 = no, 1 = yes; Math, CS, or engineering extracurricular activities: 0 = no, 1 = yes; No. of college credits completed prior to taking the course: 1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+; CS = Computer Science; CTE = Career and Technical Education

When pre-college experiences were considered to predict scores in the Impact of Computing category of the ECTD, only the number of college credits completed prior to taking the course was a significant factor, as shown in Table 11.

Table 11. Regression Coefficients of Pre-College Experiences on Impact of Computing of the ECTD (*n* = 757)

Variable	Unstandardized		Standardized	t	p	Collinearity	
	B	S.E.	β			Tolerance	VIF
(Constant)	2.39	0.14		17.47	< .001		
Higher level math courses	0.14	0.11	0.048	1.28	0.200	0.95	1.05
CS courses	0.06	0.11	0.020	0.55	0.585	0.95	1.05
Engineering or CS related CTE courses	0.03	0.10	0.012	0.32	0.749	0.96	1.04
Math, CS, or engineering extracurricular activities	-0.03	0.10	-0.013	-0.35	0.724	0.92	1.09
No. of college credits completed prior to taking the course	0.16	0.05	0.120	3.26	**0.001**	0.96	1.04

Note. $R^2 = 0.019$, Adjusted $R^2 = 0.013$; Higher level math courses: 0 = no, 1 = yes; CS courses: 0 = no, 1 = yes; Engineering or CS related CTE courses: 0 = no, 1 = yes; Math, CS, or engineering extracurricular activities: 0 = no, 1 = yes; No. of college credits completed prior to taking the course: 1 = 0-23, 2 = 24-47, 3 = 48-71, 4 = 72-92, 5 = 93+; CS = Computer Science; CTE = Career and Technical Education

Discussion

The results from this study about pre-college experience factors do not exactly support the expected finding that computer science courses taken before college would be a significant predictor for computational thinking. Prior research identified computer science courses taken during high school as a significant factor for computational thinking [3]. However, in this study, higher level computer science courses were not a significant factor in any of the multiple regressions conducted. Given that students had prior exposure to Python programming through the first introductory engineering course in Fall 2024, their prior computer science course-taking experience before college may no longer be a significant predictor.

Higher level math course taking was a significant factor for the overall ECTD score and all ECTD category scores except Impact of Computing. This finding was expected, since all the questions in the Impact of Computing category of the ECTD were word problems not requiring any computation or calculation.

The number of college credits completed prior to taking the course was by far the most significant factor in almost every multiple regression model examined. The results indicate that having more college-level courses taken, whether through AP classes, dual enrollment with community college, or other transfer credits, is beneficial for computational thinking skills. These college credits encompass all the college level courses related to a variety of subjects that each student has taken, not just those courses related to math, engineering, or computer science. This suggests that computational thinking may be more associated with college readiness than course readiness. These college-level courses, which typically cover more difficult material at a faster pace than regular high school courses, might offer students the opportunity to develop their thinking skills in general, which in turn can improve their computational thinking. In addition, having more college level courses taken might be a proxy for better-resourced high schools, since under-resourced high schools are less likely to offer AP classes and other opportunities for earning college-level credits.

Limitations and Future Research

Only 67.0% of the 1,171 students given the survey provided a valid response. This response rate might introduce sampling bias. The number of students responding to the survey with valid responses needs to be increased to reduce sampling bias.

The ECTD was also given at the end of the semester, at a point in time when students were likely to be tired from the semester and from studying for final exams. Scores on the ECTD could be lower due to a lack of effort while completing the survey. In addition, the multiple-choice guessing effect could have influenced the scores.

For future research, the ECTD can be administered twice, once at the beginning of the semester as a pre-test and once at the end of the semester as a post-test. ECTD scores between students who drop out and students who persist can be compared using the ECTD pre-test scores. The efficacy of using pre-ECTD scores as predictors for Exams can also be examined using regressions. The relationship between ECTD scores and course exam scores can be further explored by conducting structural equation modeling to test the full model between ECTD pre-test score as the predictor, course midterm exam score as the mediator, and ECTD post-test score and final exam score as the outcomes. In addition, other factors beyond pre-college experiences can be explored, including demographic factors. Demographic factors can be combined with pre-college experiences to form a more comprehensive model for ECTD scores.

Significance and Conclusion

The findings of this study revealed the extent to which various pre-college experiences influence computational thinking and identified how these factors varied across the categories of the computational thinking framework. The results of this study inform educators in designing introductory engineering courses to develop computational thinking more effectively.

References

[1] ABET. *Criteria for Accrediting Computing Programs* 2025–2026. [Online]. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2025-2026/

[2] R. J. Haddad and Y. Kalaani, “Can computational thinking predict academic performance?,” *2015 IEEE Integrated STEM Education Conference*, Mar. 2015, doi: https://doi.org/10.1109/isecon.2015.7119929.

[3] N. V. Mendoza, S. Y. Yoon, and N. G. Salvador, “Digital Equity and Computational Thinking Privilege: The Case of First-Year Engineering and Computing Students’ Attitudes towards Artificial Intelligence.,” *Computers and Education Artificial Intelligence*, vol. 9, Nov. 2025, doi: https://doi.org/10.1016/j.caeai.2025.100495.

[4] J. Wing, “Computational Thinking: What and Why?,” 2010.

[5] A. Espinal, C. Vieira, and A. J. Magana, “Assessing Disciplinary Teachers’Pedagogical and Content Knowledge in Computational Thinking,” *TechTrends*, vol. 69, no. 5, pp. 883–899, May 2025, doi: https://doi.org/10.1007/s11528-025-01072-2.

[6] M. Tissenbaum, J. Sheldon, and H. Abelson, “From computational thinking to computational action,” *Communications of the ACM*, vol. 62, no. 3, pp. 34–36, Feb. 2019, doi: https://doi.org/10.1145/3265747.

[7] J. A. Lyon and A. J. Magana, “The use of engineering model-building activities to elicit computational thinking: A design-based research study,” *Journal of Engineering Education*, vol. 110, no. 1, pp. 184–206, Jan. 2021, doi: https://doi.org/10.1002/jee.20372.

[8] J. A. Lyon, A. J. Magana, and R. A. Streveler, “Characterizing Computational Thinking in the Context of Model-Planning Activities,” *Modelling*, vol. 3, no. 3, pp. 344–358, Aug. 2022, doi: https://doi.org/10.3390/modelling3030022.

[9] A. J. Magana and G. Silva Coutinho, “Modeling and simulation practices for a computational thinking-enabled engineering workforce,” *Computer Applications in Engineering Education*, vol. 25, no. 1, pp. 62–78, Dec. 2016, doi: https://doi.org/10.1002/cae.21779.

[10] C. Vieira, M. Penmetcha, A. J. Magana, and E. Matson, “Computational Thinking as a Practice of Representation: A Proposed Learning and Assessment Framework,” *The Journal of Computational Science Education*, vol. 7, no. 1, pp. 21–30, Apr. 2016, doi: https://doi.org/10.22369/issn.2153-4136/7/1/3.

[11] J. Krauss and K. Prottsman, *Computational Thinking and Coding for Every Student: The Teacher’S Getting-Started Guide*, 1 ed. Thousand Oaks, CA, USA: Corwin, 2016.

[12] C. E. Vergara, M. Urban-Lurain, C. Dresen, T. Coxen, T. MacFarlane, K. Frazier, D. Briedis, N. Buch, A.-H. Esfahanian, L. Paquette, J. Sticklen, J. LaPrad, and T. F. Wolff, “Aligning computing education with engineering workforce computational needs: New curricular directions to improve computational thinking in engineering graduates,” in *Proc. 39th IEEE Frontiers Educ. Conf*. San Antonio, TX, USA: IEEE, Oct. 2009, pp. 1–6, doi: 10.1109/FIE.2009.5350463.

[13] College Board. *AP Computer Science Principles*. [Online]. Available: https://apstudents.collegeboard.org/courses/apcomputer-science-principles

[14] Computer Science Teachers Association. (2017). *CSTA K-12 Computer Science Standards, Revised 2017*. [Online]. Available: https://www.csteachers.org/standards

[15] K. Brennan and M. Resnick, “New frameworks for studying and assessing the development of computational thinking,” in *Proc. Annu. Meeting Amer. Educ. Res. Assoc., 2012*, pp. 1–25. Available: https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

[16] N. V. Mendoza Diaz, S. Y. Yoon, D. A. Trytten, and R. Meier, “Development and Validation of the Engineering Computational Thinking Diagnostic for Undergraduate Students,” *IEEE Access*, vol. 11, pp. 133099–133114, 2023, doi: https://doi.org/10.1109/access.2023.3335931.

[17] N. V. Mendoza Diaz, R. Meier, D. A. Trytten and S. Yoon Yoon, "Computational Thinking Growth During a First-Year Engineering Course," *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 2020, pp. 1-7, doi: 10.1109/FIE44824.2020.9274250.

[18] N. V. Mendoza Diaz, D. A. Trytten, R. Meier, and J. M. Moore, “The impact of prior programming experience on computational thinking in first-year engineering experience,” in *Proc. CoNECD (Collaborative Netw. Eng. Comput. Diversity)*, Feb. 2022, [Online]. Available: https://peer.asee.org/the-impact-of-prior-programming-experience-on-computational-thinking-in-first-yearengineering-experience