Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
ASEE
Paper ID #48854

Student Impact of First-Year Computing Blueprint

Mr. Frank Kreimendahl, Wentworth Institute of Technology

Frank Kreimendahl is an assistant professor in the School of Computing and Data Science at Wentworth Institute of Technology. He is focused on teaching computer science fundamentals and building stronger resources for student learning. He aims to bring interest and competence to algorithm-driven problem solving in the classroom.

Student Impact of First-Year Computing Blueprint

Frank Kreimendahl
School of Computing and Data Science
Wentworth Institute of Technology
Boston, MA, USA
kreimendahlf@wit.edu

***Abstract*—Course blueprints are essential for ensuring consistency and improved student outcomes in first-year computing courses. In our past work, we analyzed students' grades in first-year classes and introduced a CS I/CS II course blueprint. This paper builds on that work by measuring the impacts of that blueprint on student performance.**

Since our last paper, we have made incremental changes to the blueprint, including adding Gradescope programming labs and online compilers to increase accessibility for students from diverse backgrounds. Wentworth Institute of Technology serves inner-city Boston students who often only have access to Chromebooks. One goal is to increase the benefits realized by the blueprint for these off-campus delivered sections.

We gathered students' final grades from before and after the introduction of the blueprint. We analyze the initial impact of the blueprint's deployment on student success, measured across various demographics. In addition, we included questions related to specific course topics on student evaluations to gauge student reception of the blueprint.

***Index Terms*—computing, first-year, blueprints, CS I, CS II, student success**

I. Introduction

A. Who is Wentworth

Wentworth Institute of Technology is a 39-acre urban campus in the city of Boston where the major programs are engineering, computing, architecture, design, and management. Wentworth has approximately 4000 students who are predominantly pursuing undergraduate majors. The University has five schools: Architecture and Design, Computing and Data Science, Engineering, Management, and Sciences and Humanities. It offers 23 undergraduate degree programs and 13 graduate degree programs. Approximately 250 first-year students take CS I and CS II each year and come from majors in computing, engineering, and management.

B. What are CS I and II at Wentworth

Computer Science I (CS I) is a foundational class that is fundamental to the future success of students in computing majors at Wentworth. Other majors also rely on CS I to introduce students to the basics of programming before branching into major-specific computing courses. CS I covers the fundamental concepts and skills of programming in Java. Students learn and develop skills in problem solving, algorithm development, program design and structure, code documentation and style, and testing and debugging. Topics include data types and variables, device/file input and output, flow control and functions, use of basic data structures, as well as principles and applications of object-oriented programming.

Computer Science II (CS II) is the second foundational class offered to computing majors at Wentworth. Some other majors rely on CS II to introduce students to fundamental programming topics before branching into major-specific computing courses. CS II builds on the CS I's programming concepts and skills using Java. Students continue to develop skills in problem solving, algorithm development, program design and structure, code documentation and style, and testing and debugging. Topics include object-oriented programming, inheritance and polymorphism, GUI basics with JavaFX, abstract classes and interfaces, generics, collections, recursion, and event-driven programming.

C. Motivation and Contributions

A prior paper presented an analysis of student performance and described a blueprint designed to improve student retention through a variety of resources.[1] Since the blueprint's initial design, faculty have used it for instruction in CS I and CS II courses for six semesters. We aim to understand the impact of this change through updated analysis, and to identify aspects that can be improved further.

One of the goals of this paper is to compare student grades between pre-blueprint cohorts and post-blueprint cohorts. We previously measured the impact of many demographic differences on student outcomes. In this analysis, we focus on a few of those key measures to understand the impact that the blueprint has had on student success.

This paper also presents two other topics related to the blueprint. We present the results of a student survey on their experience with the blueprint's resources, which gives us some insight into student reception instead of relying entirely on grading outcomes to understand its impact. The paper also describes incremental updates made to the blueprint in order to support a wider range of students.

II. Literature Review

Researchers have been measuring indicators of student success in computing for almost as long as the field has existed. Many early studies focused on predicting student success with the purpose of weeding out students who were likely to fail in an environment where computing faculty and computing devices were scarce. [2], [3] As teaching resources have become more abundant and educational pedagogy has shifted, more recent studies have focused on improving the retention and performance of computing students. Many groups that are currently minorities in computer science college courses – female, non-traditional, Black, Hispanic, etc. students – have shown lower performance and retention. Numerous papers have measured these differences and sought to improve the outcomes of these students.

The literature discussed in this section focuses on the common goals of identifying why students fail first-year computing classes, and identifying resources to help them succeed. These works present methods, teaching approaches, and insights that highlight the recurring need for improved computing education.

In Newhall et al.'s work at Swarthmore, educators recognized that minority retention was statistically low compared to the rest of the student body.[4] They instituted a well-organized student mentorship program in their CS I course, and received immediate positive feedback on the additional resource. They extended mentorship to CS II when they found that it was successful in CS I. They achieved their goals of increasing minority representation, and additionally found that students proactively built study groups that led to a stronger community.

Another method used to increase performance and participation in first-year CS courses has been through the use of group projects, as presented in [5]. The authors in this paper present an innovative method of building computer science courses where all student work is done in groups, and is coupled with a summative assessment tool. The students not only solve coding tasks in groups, but also present weekly progress, helping with public speaking and presentation skills. These additional teamwork and presentation skills are important for a first-year student experience and for introductory programming. This study measured large increases in attendance rates and overall course grades increased by 22%.

The strongest support we found for our methodology in writing a comprehensive blueprint for CS I and CS II came from Cheah in [6]. They performed a literature review titled '*Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming*' and summarized one of the major factors as ineffective pedagogy. Their literature review of pedagogy found that the sources of ineffective teaching include confusing or incomplete teaching materials, ineffective teaching strategies, and improper learning outcomes.

Several other papers have measured that student performance in CS I is one of the strongest predictors of success in CS II.[7], [8] Malla et al.'s work uses a similar approach to our previous work in identifying which students have lower passing rates. Identifying the predictors of failure gives educators an opportunity to design and provide resources in a more targeted way to improve student success for everyone in the classes.

Based on these prior findings, we find strong motivation in the measurement of performance and the design of first-year computing resources. The ideas in the related works help inform our approach and provide opportunities for further pedagogical design.

III. Methodology

Without a consistent framework for CS I, many students will be unprepared for their subsequent computing courses. The course blueprint that we designed, along with the supporting documents, provides the framework for teaching the course in a consistent manner across all sections. The blueprint includes descriptions of the required tools, textbook, topics, assignments, exams, checkpoint questions for faculty and students, and a sample schedule for the course. Instructors at Wentworth use the information contained in the blueprint when forming their teaching plan. The blueprint was completed and first employed in classrooms in Fall 2021.

In this section we will describe the methodology used to understand the impact of the blueprint, as well as presenting the data from student performance. We collected data on students' course grades from Fall 2016 to Spring 2024. The data is split into two time ranges – student performance before the blueprint was introduced (2016-2020) and after (2021-2023). This data shows the performance of our students in first-year computing courses compared across many different demographics.

A. Rationale

The rationale behind this research is three-fold:

1) Increase student retention at our institution
2) Provide a transformative student experience, which is one of our four strategic pillars
3) Provide a scaffold to our faculty while providing flexibility of pedagogy

B. Methodology

- Compare student grades before and after the deployment of the blueprint.
- Identify changes that the introduction of the blueprint may have caused, based on comparisons.
- Incorporate improvements to some sections of the initial blueprint.
- Understand the student experience with specific parts of the blueprint.

C. Performance Data

The data collected show students compared by several demographic measures. The sample size of students in the 2021-2023 time span consisted of 589 students who attempted CS I. Data analysis focused on each student's first attempt at a course, as our educational goal is for students to pass a course the first time they take it. We did not include data on student attempts to pass CS I or CS II after failing the first time.

The results of each measure are split into a pair of tables – the first table shows measurements from before the blueprint was deployed and the second table shows measurements from after the blueprint was deployed.

Tables I and II show the pass rate of the students in CS I and CS II by year and gender. The tables show that there

TABLE I
CS I AND CS II PASS RATES BY GENDER (2016-2020)

Gender	2016	2017	2018	2019	2020	Average
Female	89%	87%	90%	83%	89%	88%
Male	88%	89%	89%	85%	89%	88%

TABLE II
CS I AND CS II PASS RATES BY GENDER (2021-2023)

Gender	2021	2022	2023	Average
Female	83%	93%	89%	88%
Male	88%	88%	89%	88%

continues to be a high rate of success in these courses and that there has not been a measurable change in performance since the blueprints were introduced in 2021. About 12% of the students counted are female students.

TABLE III
CS I AND CS II PASS RATES BY RACE/ETHNICITY (2016-2020)

Race/Ethnicity	2016	2017	2018	2019	2020	Average
Asian	89%	86%	92%	86%	88%	89%
Black or African American				67%	70%	72%
Hispanic		86%	73%	77%	83%	80%
White	91%	90%	91%	88%	92%	91%
Average	88%	88%	89%	84%	89%	88%

Blank cells indicate sample size is too low.
Blank cell results are included in averages.

TABLE IV
CS I AND CS II PASS RATES BY RACE/ETHNICITY (2021-2023)

Race/Ethnicity	2021	2022	2023	Average
Asian	94%	88%	97%	93%
Black or African American	65%	75%	79%	73%
Hispanic	69%	89%	75%	77%
White	90%	90%	93%	91%
Average	87%	89%	89%	88%

Tables III and IV show the pass rates of our students based on race and ethnicity. Wentworth consists of 56% white students and all other races and ethnicities constitute the remaining 42%.

Comparing the data in these two tables, we see that the recent averages have not shifted considerably from the pre-blueprint averages. White students performed similarly before and after the blueprint's introduction. Asian students were already performing well, and saw an increase in pass rates in recent years. Black or African American students consistently have below average pass rates, but show a clear trend in improvement over time. Hispanic students, however, had lower pass rates average pass rates after the blueprint, highlighting a demographic to target with additional scaffolding. Two of the lowest pass rates in either table appear in the Hispanic and Black student populations in 2021.

TABLE V
CS I VS. CS II GRADE RELATIONSHIPS (2016-2020)

CS I Outcome	CS II Outcome (2016-2020)		
	Fail	Pass, C or lower	Pass, higher Than C
Pass, C or lower	15%	38%	47%
Pass, higher than C	3%	16%	81%

18% of students are required to take CS I but not CS II.

TABLE VI
CS I VS. CS II GRADE RELATIONSHIPS (2021-2023)

CS I Outcome	CS II Outcome (2021-2023)		
	Fail	Pass, C or lower	Pass, higher Than C
Pass, C or lower	25%	35%	40%
Pass, higher than C	10%	14%	76%

Tables V and VI show the relationship of student performance between CS I and CS II across the two measured time periods. After the blueprint was introduced, students who passed CS I on their first attempt failed CS II more frequently. Students who passed CS I in both categories – C or lower, and above a C – failed CS II at much higher rates. 25% of students who passed CS I with a C or below failed CS II, whereas only 10% of students who passed CS I with a grade above a C failed CS II.

D. CS I Blueprint Changes

The blueprint consists of all the necessary resources for an instructor to deliver the class. The CS I blueprint has been changed since its deployment in 2021 to improve some of its resources.

All of the lectures, in-class labs, and programming assignments in the blueprint were rewritten. The changes did not fundamentally change the lecture topics, but improved the explanations, examples, and visual elements of the learning materials. Similarly, changes to programming labs and assignments did not fundamentally change the goals or outcomes, but improved the clarity of the assignments and simplified the explanations of some tasks.

Additionally, the framework for student work was modified. Wentworth used to provide every student with a laptop, which guaranteed that all students had access to the same computing resources. The laptop program was discontinued and the course has been updated to be more easily delivered to students with access to a range of computing devices. The blueprint now provides directions to use Gradescope and an online Java compiler, which guarantees that all students have access to the same Java environment.

E. Student Survey

In Fall 2024, additional questions were included in CS I's and CS II's student evaluations. These questions were written in order to understand student opinions on the pacing of individual topics in the courses. The questions cover all of the major topics for both courses. The responses are valuable to understand the impact of the blueprint, and show with more granularity which parts can be improved.

102 CS I students submitted responses to the questions, and 28 CS II students submitted responses. The CS II sample size is much smaller because the evaluations were collected in the fall semester, and most students who take CS II do so in the spring.

The student responses in Table VII show that the majority of CS I students are comfortable with the amount of time spent on each topic in the course. At most 3% of responding students thought that any topic is moving too slowly. Many more students felt that topics were covered too quickly, particularly in the categories of Expressions, Testing & Debugging, Exceptions, File I/O, and ArrayLists. Almost 20% of the respondents did not remember their experience with OOP.

Table VIII shows the major topics covered in CS II, along with student responses to each topic. The responses show that, with a few exceptions, the largest category of students liked the pace of each topic. The three exceptions, GUI programming, data structures, and event-driven programming, show that many students wanted to spend more time on those topics. The Object Oriented Programming topic stands out as the only topic with more than one student responding that the material was covered for too long.

IV. Discussion

In this section, we analyze and discuss the results of the performance data and student survey. We make a number of observations between the pre-blueprint grades and post-blueprint grades that help us understand what impact the blueprint may have had, and that highlight areas where the blueprint may be improved further. Analyzing the results of the student survey gives us additional opportunities to understand the current delivery of CS I and CS II, and how it may be changed to benefit students.

A. Pass Rates and Grade Comparisons

In both tables I and II, we see a strong parity between female and male students. There is higher yearly variance in female performance, which may be attributed to the smaller sample size. Only 11-12% of the students taking CS I and CS II courses are female. Two conclusions that we draw from these tables are that targeting a specific gender in our blueprint's scaffolding is unnecessary, and that there is not a measurable difference in overall student grades since the blueprints have been introduced.

Comparing changes between III and IV reveals interesting differences between the two time periods. In 2021, both Black and Hispanic students had very low pass rates. These outliers may be explained by impacts of the COVID-19 pandemic, which had different effects depending on race/ethnicity. White and Asian students have performed well through both time periods. Focusing on the Black or African American achievement over time, pass rates increase every year except 2021. The original design for the blueprint included specific scaffolding tools to better support students of color, so this continued improvement is encouraging.

Hispanic students had an atypically low year in 2021 and an atypically high year in 2022, but their post-blueprint average was the only group to decrease. This presents an opportunity to build extra support for Hispanic students in introductory computing classes.

In tables V and VI, we compare students who took both CS I and CS II in the two separate time periods. Both tables show that students who perform poorly in CS I are more likely to fail or perform poorly in CS II. In the last three years, students who passed CS I the first time failed CS II more frequently. Fully 25% of students who achieved a C or lower in CS I failed CS II, up from 15%. This may be an unintended consequence of the blueprint, showing an effect where the CS II materials assume that students have mastered the CS I topics, and materials between the two courses are more tightly coupled. If a student passes CS I with a weak understanding of some foundational topics, CS II becomes more difficult.

These increases in failure rates make it clear that CS I concepts are foundational for the rest of students' computing education, and that early scaffolding can have a large impact on student retention and future success. Improvements to the CS I blueprint can improve student learning beyond that single course.

B. Student Responses

The responses to the student evaluations provide valuable information that could not be derived from the grade analysis alone. With questions that focus on specific topics in the CS I and CS II curricula, we can better understand students' experience with the blueprint.

In table VII, very few students reported that any topic moved too slowly. For every topic, the majority of students liked the pacing of instruction. Object-oriented programming is a notable outlier in the 'Don't Remember' column. This may indicate that students do not clearly understand what OOP concepts entail, and therefore cannot recall the pacing of that topic.

Focusing on the 'Too Short' column, three topics stand out as moving too quickly: "Expressions, Testing, and Debugging", "Exceptions and File I/O", and "ArrayLists". Students may struggle with early testing and debugging topics because many of them are learning computational concepts for the first time, and testing and debugging strategies often require a higher cognitive skill level than programming topics like conditional statements or loops. The topics at the end of the semester probably appear too short because some sections rushed through them before the final exam period, and students had no opportunities to reinforce the concepts once they were introduced to them.

TABLE VII
CS I STUDENT RESPONSES

Topic	Don't Remember	Too Short	Just Right	Too Long
Variables, I/O, Types, Strings	10	12	77	3
Control Flow, Conditionals	9	9	82	2
Expressions, Testing & Debugging	9	28	64	1
Loops	10	15	77	0
Arrays	5	19	76	2
Object Oriented Programming	19	23	60	0
Exceptions, File I/O	9	27	64	2
ArrayLists	5	30	66	1

TABLE VIII
CS II STUDENT RESPONSES

Topic	Don't Remember	Too Short	Just Right	Too Long
Object Oriented Programming	4	3	16	5
Inheritance and Polymorphism	7	2	19	0
GUI Programming	7	15	6	0
Abstract Classes, Interfaces	5	2	20	1
Generics, Collections, Iterators	7	7	13	1
Lists, Stacks, Queues, Sets, Maps	5	10	13	0
Recursion	9	6	13	0
Event-Driven Programming	8	10	9	1

Table VIII shows the course evaluation responses from 28 CS II students. A few students retained CS I OOP information enough that they thought the OOP review was too slow, but there were no indicators that other topics moved too slowly. Three topics stand out in the 'Too Short' column, including a majority of students wanting more focus on GUI programming. This is a clear signal to revise the blueprint to bolster the GUI programming topic. The students also felt that they did not have enough time with the data structures topic. This topic is meant as a brief introduction to common data structures and many students take a dedicated Data Structures course that examines the topic in much more depth. Two topics at the end of the semester also appear rushed.

V. SUMMARY

In this paper, we extend our research that aims to improve student success in CS I and CS II courses. We present and analyze grade data to compare student success rates before and after the introduction of the blueprint. We also present incremental changes to our original blueprint. Lastly, we share results of student surveys on individual topics within the courses to better understand their experience with the delivery of blueprint materials.

In our analysis of grades, we found that performance between genders has continued to be equal. Comparing changes in performance between race/ethnicity groups, we saw that Black student pass rates have modestly improved, and Hispanic students also fell below the average every year but did not show consistent yearly improvement. This shows that there are still opportunities to improve our blueprint to support student success. We previously measured that low grades in CS I led to lower grade outcomes in CS II, and that trend has strengthened. This may be attributed to a tighter coupling of topics between CS I and CS II, leading to students who have low comprehension of CS I topics missing opportunities in CS II to build on their earlier knowledge.

Analyzing student evaluations, we found several insightful patterns. Students did not feel like the pace of either course was too slow. Students felt like they spent too little time learning about topics that appeared at the end of the course, and topics that introduced the basics of large concepts like event-driven programming and debugging.

FUTURE WORK

This paper and our continuing work lead to many interesting questions and analyses that we plan to pursue.

A. *CS I Group Project*

Many of the papers in our literature review discussed the advantage of well-designed group projects in CS I. The authors increased success in retention and persistence rate. We will create a group project in CS I and study the impact in terms of student reflections and course learning outcomes.

B. *CS II Blueprints Description and Review*

We plan to write a paper that describes the details of the CS II blueprint that we have built, which follows a similar pedagogical approach to the CS I blueprint. The CS II blueprint includes similar faculty and student resources to ensure a consistent and successful approach to teaching the full first year of computing courses.

C. *Second-Year Computing Blueprints*

Once we have built a strong framework for the first-year computing courses, we plan to build similar blueprints for our second year of fundamental computer science courses, including two courses: Data Structures and Algorithms. We hope to extend the benefits of a blueprint to second-year students.

D. Student Tutoring

Although we already have undergraduate tutors who offer help in CS I and CS II, we hope to enhance their positive impact through increased visibility in the classroom. We want to study the relationship between tutoring attendance and success in first-year computing and potentially change the structure of our student tutoring sessions based on the results.

ACKNOWLEDGMENT

We would like to thank Micah Schuster for his work on the blueprints and Steven Sherrin for collecting the data that was essential for our analysis of student demographics and performance.

REFERENCES

[1] F. Kreimendahl and D. Suresh-Menon, "Creating a blueprint for success in first-year computing," in *2023 ASEE Annual Conference & Exposition*, 2023.

[2] L. H. Werth, "Predicting student performance in a beginning computer science class," *SIGCSE Bull.*, vol. 18, no. 1, pp. 138–143, Feb. 1986.

[3] D. F. Butcher and W. A. Muth, "Predicting performance in an introductory computer science course," *Commun. ACM*, vol. 28, no. 3, pp. 263–268, Mar. 1985.

[4] T. Newhall, L. Meeden, A. Danner, A. Soni, F. Ruiz, and R. Wicentowski, "A support program for introductory cs courses that improves student performance and retains students from underrepresented groups," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '14, Atlanta, Georgia, USA: Association for Computing Machinery, 2014, pp. 433–438.

[5] S. Billings and M. England, "First year computer science projects at coventry university: Activity-led integrative team projects with continuous assessment," in *Proceedings of the 4th Conference on Computing Education Practice*, ser. CEP '20, Durham, United Kingdom: Association for Computing Machinery, 2020.

[6] C. S. Cheah, "Factors contributing to the difficulties in teaching and learning of computer programming: A literature review," *Contemporary Educational Technology*, vol. 12, no. 2, ep272, 2020.

[7] S. Malla, J. Wang, W. Hendrix, and K. Christensen, "Predicting success for computer science students in cs2 using grades in previous courses," in *2019 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2019, pp. 1–5.

[8] A. N. Kumar, "Predicting student success in computer science–a reproducibility study," in *2018 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2018, pp. 1–6.