IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15749

Implementing Culturally Responsive Computer Science in Higher Education: A Framework for Code, Culture, and Classroom

Dr. Sarika H. Gadekar[1], Dr. Vidya S. Chandgude[2], Dr. Kanchan G. Rathi[3], Mrs. Shivani P. Mahajan[4]

Assistant Professor, Department of Computer Application, MAEER's MIT ACSC Alandi, Pune, India[1-4]

Abstract: Although the demand for a diverse talent base has increased in the global technology industry, introductory Computer Science courses in higher education still have high dropout rates, especially for female and minority students. Traditional Computer Science courses give abstract, hard programming problems irrelevant to students' lives and social norms. This paper proposes a practical framework for the design and implementation of Culturally Responsive Computer Science Curricula (CRCSC) in higher education. The framework actively brings culture into computational thinking by introducing "civic context" through community-based projects that incorporate locally available datasets to solve civic problems (i.e., the possibility of using datasets from the community to address civic inequities). This paper describes a mixed-methods case study of this framework for an introductory programming course (CS1). Quantitative assessment results indicate a significant increase in student self-efficacy and a decrease in course attrition. Most importantly, the overall course dropout rate dropped from 21.4% (historical average) to 6.1%, with high gains for female and minoritized students. The paper summarizes practical tips and concrete pedagogical mappings for university teachers focusing on updating Computer Science curricula to design a more inclusive talent ecosystem.

Keywords: Computer Science, Education, Pedagogy, Diversity, STEM, Curriculum Design, Student Retention.

I. INTRODUCTION

One of the most essential fields of academic study and professional advancement is Computer Science (CS). Yet, despite decades of effort, the college student body remains far from demographic diversity compared to the general population. Plenty of ink has been spilled on the "leaky pipeline" in computer science; although there have been small increases in enrollment rates of underrepresented groups (female, Black, Hispanic, and Indigenous students), their attrition from introductory courses remains disproportionately high.

Research on this retention gap has historically focused on deficit models, suggesting that students from minoritized backgrounds arrive at universities underprepared, logically fragile, or lacking intrinsic interest [1], [9]. More recent work, however, has moved away from attitudes of student deficits towards identifying institutional and pedagogical barriers. CS1 and CS2 sequences are often designed as gatekeeper or "weed-out" courses. These classes have traditionally been geared towards students who come with prior informal programming knowledge, reinforcing an overwhelmingly homogenous demographic [6], [8].

Traditional computer science curricula often suffer from what educators refer to as "epistemic distance." Even foundational programming concepts are almost always taught using abstract mathematical puzzles (such as finding the N-th Fibonacci number, sorting a sequence of random integers, or building text-based card games). Though these practices isolate computational logic effectively, they do not demonstrate how software engineering can serve as a mechanism for community empowerment, social justice, or humanistic inquiry [17], [18]. These practices are particularly distancing to female and minority students, who research shows frequently prefer career paths with well-defined social utility, collaborative environments, and community impact [11], [13].

This paper addresses these structural inequities and presents a robust framework: the Culturally Responsive Computer Science Curricula (CRCSC) model for higher education. Grounded in Culturally Responsive Pedagogy (CRP) and Culturally Sustaining Pedagogy (CSP) originally formulated for K-12 education [1], [2], [3], we modify these practices to match the rigor of university-level software engineering courses.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15749

The contributions of this work are threefold:
1. We introduce a concrete, rubric-based curricular mapping model that transforms traditional coding assignments into realistic, contextually rich coding tasks relevant to social issues without sacrificing technical rigor.
2. We perform a mixed-methods empirical assessment of the framework applied to N = 114 undergraduate students.
3. We demonstrate that culturally responsive pedagogy leads to statistically significant gains in student self-efficacy, retention, and a sense of belonging.

II. LITERATURE REVIEW

A. The Attrition Crisis in Introductory Computing

Globally, attrition rates in CS1 courses range between 20% and 40% [6], [8]. Research indicates that this dropout rate is not strictly related to technical inability; instead, it is heavily connected with psychological variables, primarily academic self-efficacy (beliefs about one's ability to successfully perform academic tasks) and feelings of fit within the discipline [7], [10].

In a classroom environment that does not reflect them physically or historically, underrepresented students experience high levels of impostor syndrome, particularly when course materials assume prior technical literacy [11], [12]. This isolation is exacerbated for female and minority students by traditional competitive classroom structures that discourage collaboration.

B. Culturally Responsive Pedagogy (CRP) and Its Evolution

Culturally Responsive Pedagogy, developed by Ladson-Billings [1] and elaborated by Gay [2] and Paris [3], is an asset-model approach to instruction. Instead of erasing or ignoring students' cultural backgrounds, it leverages them as vehicles for learning. In computer science, early foundational work on Culturally Responsive Computing (CRC) established that tech education is not culturally neutral [4], [5]. Algorithms, user interfaces, and software architectures inherently reflect the biases and values of their creators.

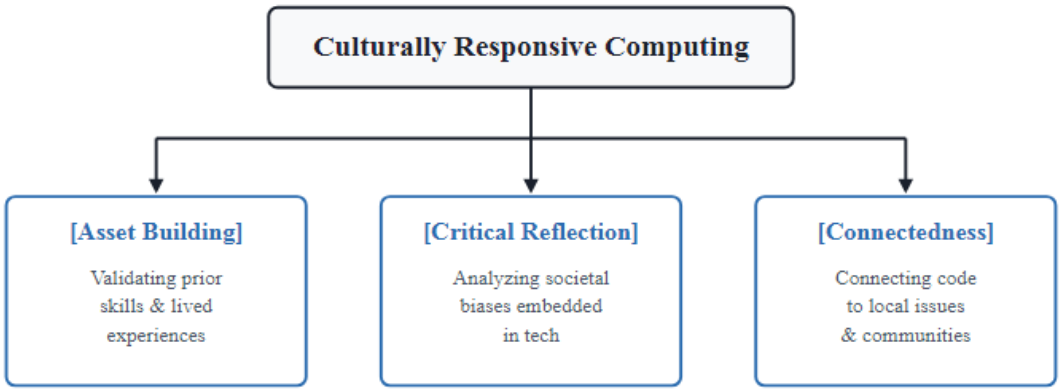

Figure 1 illustrates the core pillars of CRC: Asset Building, Critical Reflection, and Connectedness.

Curricula should incorporate three main components to ensure computing is highly responsive:
Asset Building: Identifying and validating students' prior non-traditional technological skills and lived experiences.
Critical Reflection: Encouraging students to analyze how technology interacts with structures of social power and systemic bias.
Connectedness: Linking computational problems directly to students' local communities and personal identities.
This paper disproves the assumption that cultural relevance dilutes academic severity, demonstrating that cultural context and technical complexity are highly complementary [17], [18].

III. THE CRCSC FRAMEWORK

The proposed Culturally Responsive Computer Science Curricula (CRCSC) framework relies on the concept of computational translation. We do not alter fundamental computer science concepts (e.g., variables, control flow, object-oriented design, or algorithm analysis); rather, we translate the context in which these concepts are introduced, practiced, and evaluated.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15749

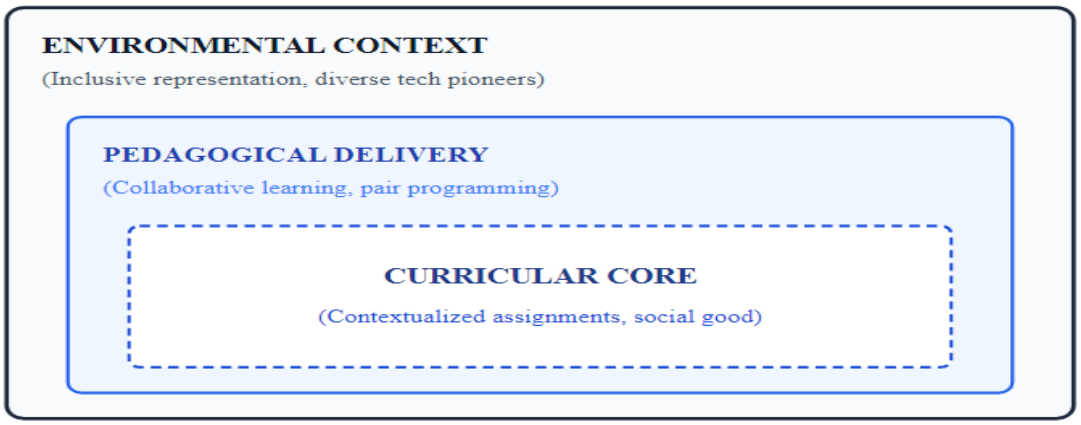

Fig. 2. Three-layer structure of the CRCSC Framework.

A. The Curricular Core: Asset-Based Assignments

The core of our framework replaces sterile programming prompts with projects requiring engagement with real-world, civic, and cultural datasets. This shifts the student's identity from a passive code consumer to an active agent of social change. Table I demonstrates this assignment transformation.

TABLE I: ASSIGNMENT TRANSFORMATION MAPPING

Core CS Concept	Traditional Assignment	Culturally Responsive Alternative	Societal / Cultural Focus
Variables, Conditionals	Calculate compound interest or simulate retail checkout.	Environmental Justice Index: Calculate local exposure to air pollutants from regional census data.	Analyzing environmental inequalities and public health correlations.
Loops, Iteration, Arrays	Sort a list of 10,000 randomly generated integers.	Civic Transit Inequity Tracker: Filter and sort municipal transit datasets for bus delays in redlined areas.	Investigating spatial justice and urban planning disparity.
Functions, State Mgmt.	Build a text-based console game (Blackjack / Tic-Tac-Toe).	Oral History Database: Build an interactive narrative engine that preserves community stories.	Preserving indigenous knowledge, linguistic diversity, and oral history.
Databases, Queries	Write SQL queries to extract corporate employee salaries.	Food Desert Analytics: Query economic databases to identify gaps in fresh food access by zip code.	Highlighting socioeconomic resource allocation and food insecurity.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15749

B. Pedagogical Delivery: Agile and Collaborative Learning

Traditional instruction often presents coding as a competitive, solitary activity, alienating students outside dominant CS demographics. Our framework replaces this model with structured Cooperative Learning Protocols, primarily formal Pair Programming based on industry Agile practices [14], [15], [16].

Students are assigned using heterogeneous grouping strategies that balance varied levels of prior experience. Partners rotate bi-weekly to prevent cliques and build an integrated peer network. Grading schemes explicitly reward group code documentation and peer code reviews.

C. Environmental Context: Inclusion Guarantee

The final layer addresses visual and historical representation in the classroom. Historical narratives highlighting underrepresented computing pioneers introduce theoretical concepts throughout the semester:
Compilation & High-Level Languages: Highlighted through Admiral Grace Hopper.
Trajectory Calculation & Databases: Framed around the mathematical contributions of Katherine Johnson and Gladys West.
Algorithmic Sequences & Early Design: Framed using Ada Lovelace, ancient non-Western counting systems (such as the Inca Quipu), and the binary logic of the I Ching.

IV.METHODOLOGY

We performed a quasi-experimental, mixed-methods study evaluating the CRCSC framework during a 15-week academic semester in an introductory computer science course (CS1) at a large public university.

A. Participant Cohort Demographics

The study included N = 114 undergraduate students across three lecture sections. Self-reported demographics are summarized in Table II.

TABLE II: DEMOGRAPHIC BREAKDOWN OF THE STUDENT COHORT (N = 114)

Demographic Variable	Category	Count (n)	Percentage (%)
Gender Identity	Female	48	42.10%
	Male	62	54.40%
	Non-binary / Other	4	3.50%
Academic Major	Computer Science / Software Eng.	38	33.30%
	Other STEM (Eng., Math, Bio.)	51	44.70%
	Non-STEM / Undecided	25	22.00%
Prior Experience	None (Absolute Beginner)	72	63.20%
	Minimal (Self-taught / High School)	31	27.20%
	Moderate (Familiar with ≥1 language)	11	9.60%

B. Educational Intervention Setup

The CRCSC framework completely replaced the traditional CS1 curriculum over 15 weeks. The implementation focused on Contextualized Assignments, Collaborative Pair Programming, and Inclusive Historical Anchoring.

C. Survey Instrumentation and Administration

The 10-item CS Self-Efficacy and Belonging Survey (CS-SEBS) measured Computing Self-Efficacy (SE) and Sense of Belonging (SB) on a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree). Surveys were administered pre-course (T1) and post-course (T2). Table III details the items.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15749

TABLE III: CS-SEBS INSTRUMENT SURVEY ITEMS

Subscale	Item ID	Survey Question Statement
Self-Efficacy (SE)	**Q1**	I am confident in my ability to write computer programs to solve complex, real-world problems.
	Q2	When I get stuck on a difficult bug, I feel confident I can find a way to fix it.
	Q3	I believe I have the logical skills required to succeed in a computer science major.
	Q4	I can explain basic CS concepts (loops, conditionals, arrays) to a peer.
	Q5	I feel comfortable designing an algorithm from scratch for a novel problem.
Belonging (SB)	**Q6**	I feel like a valued and accepted member of this computer science classroom.
	Q7	I see a clear connection between coding assignments and issues that matter to my community.
	Q8	People like me are represented in the examples, history, and leadership in this course.
	Q9	I feel comfortable asking questions or collaborating with peers during lab sessions.
	Q10	I can easily envision myself pursuing a long-term career as a software engineer or computer scientist.

D. Statistical Analysis Plan

Quantitative data was analyzed using Python: Reliability Analysis (Cronbach's alpha), Descriptive Statistics (mean scores, standard deviations), Inferential Hypothesis Testing (paired-samples t-test with $p < 0.05$ threshold), and Effect Size Calculation (Cohen's d).

V. RESULTS

A. Reliability Analysis of Instrument

Cronbach's alpha confirmed high internal consistency for both subscales: Computing Self-Efficacy (alpha = 0.86) and Sense of Belonging (alpha = 0.89).

B. Changes in Self-Efficacy and Belonging Quantitatively

Statistically significant positive gains were observed across all ten survey items from pre-test (T1) to post-test (T2), detailed in Table IV.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15749

TABLE IV: PRE- AND POST-COURSE SURVEY ITEM MEAN SCORES (N = 114)

Item	Pre (T1)	Post (T2)	Mean Δ	t-stat	p-value	Cohen's d
Q1	2.31 (0.88)	4.22 (0.65)	+1.91	18.42	<0.001	1.72
Q2	2.14 (0.91)	3.98 (0.74)	+1.84	16.11	<0.001	1.51
Q3	3.02 (1.04)	4.15 (0.68)	+1.13	9.85	<0.001	0.92
Q4	2.55 (0.97)	4.41 (0.51)	+1.86	17.52	<0.001	1.64
Q5	1.98 (0.82)	3.89 (0.79)	+1.91	19.10	<0.001	1.79
Q6	2.45 (0.95)	4.34 (0.62)	+1.89	16.94	<0.001	1.59
Q7	1.82 (0.79)	4.58 (0.53)	+2.76	28.31	<0.001	2.65
Q8	2.01 (1.12)	4.27 (0.68)	+2.26	19.45	<0.001	1.82
Q9	2.88 (1.05)	4.52 (0.55)	+1.64	14.22	<0.001	1.33
Q10	2.41 (1.18)	4.10 (0.81)	+1.69	12.38	<0.001	1.16

The largest gain occurred in Q7 (Community Connection) with an increase of +2.76 ($t = 28.31$, $p < 0.001$, Cohen's $d = 2.65$). Aggregated Self-Efficacy increased from 2.40 to 4.13 ($p < 0.001$) and Sense of Belonging increased from 2.31 to 4.36 ($p < 0.001$).

C. Course Completion and Academic Performance

The overall course DFW rate (D/F grades or Withdrawals) was reduced from a historical baseline of 21.4% to 6.1% (7/114 students). Female withdrawal dropped from 24.5% to 4.2% (2/48 students). Final exam performance was maintained (CRCSC Mean = 83.4%, SD = 8.2% vs Historical Mean = 81.9%, SD = 10.1%), proving that academic rigor was fully preserved.

VI. DISCUSSION & CONCLUSION

The empirical findings illustrate that contextualizing computational problems within real-world civic frameworks significantly enhances student retention and self-efficacy without compromising technical standards. Integrating culturally responsive assignments, collaborative pair programming, and inclusive historical contexts mitigates isolation and creates an accessible entry point into computer science for diverse student demographics.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15749

REFERENCES

[1]. G. Ladson-Billings, "Toward a theory of culturally relevant pedagogy," American Educational Research Journal, vol. 32, no. 3, pp. 465–491, 1995.
[2]. G. Gay, Culturally Responsive Teaching: Theory, Research, and Practice, 3rd ed. New York, NY: Teachers College Press, 2018.
[3]. D. Paris, "Culturally sustaining pedagogy: A needed change in stance, terminology, and practice," Educational Researcher, vol. 41, no. 3, pp. 93–97, 2012.
[4]. K. A. Scott, K. Sheridan, and K. Leyva, "Culturally responsive computing in practice: Understanding how culturally responsive computing can foster equity in CS education," Communications of the ACM, vol. 60, no. 4, pp. 88–95, 2017.
[5]. S. Sentance et al., "Culturally responsive teaching in computing: A framework for educator development," ACM Transactions on Computing Education, vol. 22, no. 2, pp. 1–28, 2022.
[6]. J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," ACM SIGCSE Bulletin, vol. 39, no. 2, pp. 32–36, 2007.
[7]. K. Quille and S. Bergin, "CS1: How will they do? How can we help? A decade of research and practice," Computer Science Education, vol. 29, no. 2-3, pp. 254–282, 2019.
[8]. C. Watson and F. W. Li, "Failure rates in introductory programming revisited," in Proc. 2014 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2014, pp. 39–44.
[9]. J. Margolis and A. Fisher, Unlocking the Clubhouse: Women in Computing, Cambridge, MA: MIT Press, 2002.
[10]. A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change," Psychological Review, vol. 84, no. 2, pp. 191–215, 1977.
[11]. A. Master, S. Cheryan, and A. N. Meltzoff, "Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science," Journal of Educational Psychology, vol. 108, no. 3, pp. 424–437, 2016.
[12]. S. Cheryan, A. Master, and A. N. Meltzoff, "Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes," Frontiers in Psychology, vol. 6, art. 49, 2015.
[13]. R. M. Marra, K. A. Rodgers, D. Shen, and B. Bogue, "Women engineering students: Self-efficacy, a sense of belonging, and persistence," Journal of Engineering Education, vol. 98, no. 1, pp. 27–38, 2009.
[14]. N. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "Pair programming improves student retention and major persistence in computer science," Communications of the ACM, vol. 49, no. 8, pp. 90–95, 2006.
[15]. L. A. Williams and R. R. Kessler, "All I really need to know about pair programming I learned in kindergarten," Communications of the ACM, vol. 43, no. 5, pp. 108–114, 2000.
[16]. B. Hanks, S. Fitzgerald, R. McCauley, L. Murphy, and C. Zander, "Pair programming in education: A literature review," Computer Science Education, vol. 21, no. 2, pp. 135–173, 2011.
[17]. M. Guzdial, "Contextualized computing education," Computer, vol. 43, no. 5, pp. 110–112, 2010.
[18]. A. T. Cross et al., "Computing for social good in introductory computer science: Impact on student engagement and diversity," ACM Transactions on Computing Education, vol. 21, no. 4, pp. 1–25, 2021.

