IJCSES

Volume 7, No 2
September 2025
ISSN 2513-8359

International Journal of Computer Science Education in Schools

Editors

Dr Filiz Kalelioglu

Dr Yasemin Allsop

www.ijcses.org

International Journal of Computer Science Education in Schools

September 2025, Vol 7, No 2

DOI: 10.21585/ijcses.v7i2

Table of Contents

International Journal of Computer Science Education in Schools, September 2025, Vol. 7, No. 2
1ISSN 2513-8359

The Middle-to-High School Transition:
Key Factors Shaping 9th-Grade Computer Science Enrollment

David J. Amiel & Cynthia L. Blitz
Center for Effective School Practices, Rutgers University, USA

DOI: 10.21585/ijcses.v7i2.234

Abstract:

The increasing demand for computer science (CS) skills underscores the importance of integrating CS education into K–12 curricula to best prepare students for a digitally-driven society. Despite significant progress in expanding access to CS courses, disparities in participation persist, especially among historically underrepresented groups. This study examines the transition from 8th to 9th grade (occurring in the U.S. around age 15) as a pivotal juncture in CS education, analyzing factors linked to 9th-grade CS course-taking among 5,505 students across eight diverse school districts in a northeastern state of the U.S. using logistic regression. Findings show that high academic achievers, male students, Asian students, and those with exposure to CS and Algebra 1 in middle school were more likely to enroll in 9th-grade CS courses. Conversely, participation is lower for females, English Language Learners, and students receiving special education services. These results point to persistent barriers to CS participation extending beyond access alone. We discuss practical implications for middle and high schools, emphasizing the need for targeted outreach and early exposure to CS to foster a sense of belonging and applicability of CS. By identifying actionable strategies to address participation gaps, this study provides data-driven recommendations for advancing equity in CS education during the critical middle-to-high school transition.

Keywords: computer science education, K-12 STEM education, broadening participation in computing, secondary education, STEM course-taking

1. Introduction

Jobs in computing and related fields are experiencing rapid growth (US Bureau of Labor Statistics, 2024), highlighting the continued and increasing demand for computer science (CS) skills. At the same time, as technology, artificial intelligence (AI), and automation permeate nearly every sector of society, these skills are no longer relevant only for those pursuing careers in software development or technical fields but are seen as critical for all students to become responsible, informed, and productive members of society. Thus, meaningfully incorporating computer science education (CSE) into K-12 curricula continues to gain momentum as a national priority. Recent efforts in computer science education (CSE) are focused on both addressing persistent participation gaps among underrepresented groups (CSTA et al., 2024; NASEM, 2024; TeachAI & CSTA, 2024) as well as promoting more universal participation in CSE (Barr, 2022; Li et al., 2020; Michaeli et al., 2023). Broadening participation in CSE is paramount, ensuring that *all* students not only have access to CSE, but participate as well.

Over the past decade in the United States, significant investments have been made in K-12 CSE at the local, state, and federal levels. Notable initiatives, such as Computer Science for All, the Computer Science Education Act, the Future of Work, and Broadening Participation in Computing, have contributed to progress in expanding CSE. For instance, while only 45% of public high schools offered foundational computer science in 2019, this figure rose to 60% by 2024 (Code.org, 2024). Although access has improved, it is not yet universal (Fletcher & Warner, 2021; NCES, 2019), and it remains skewed based on student demographics and locale. Rural, urban, and smaller high schools, as well as those with higher percentages of economically disadvantaged students, are less likely to offer foundational CS (Code.org, 2024). Further, U.S. K-12 students that are Black/African American, Hispanic/Latinx, or Native American/Alaskan are less likely to attend schools that offer such courses (Code.org, 2024).

We know, however, that increasing CSE *access* does not imply increases in *participation* (Adrion et al., 2020; Code.org, CSTA et al., 2024, ECEP Alliance, 2024; Warner et al., 2021), a dimension to broadening participation efforts that has been discussed for over a decade (Margolis et al., 2012). In the 2023-2024 school year, only 6.4% of high school (HS) students in the U.S. enrolled in a foundational CS course. In the U.S. CS courses are typically offered as opt-in electives, and specifics on elective processes vary across schools. Despite ongoing efforts to encourage participation, males are still twice as likely to enroll in these courses as females, a statistic that has remained unchanged over the past four years (Code.org, 2024).

Participation among racial and ethnic groups has also been relatively stable over this period: broadly speaking, Black/African American, Hispanic/Latinx, Native Hawaiian/Pacific Islander, multiracial, English Language Learners (ELLs), students with disabilities, and economically disadvantaged students remain underrepresented in their CSE participation (Code.org, 2024). For example, while 53% of students nationally qualify for free and reduced lunch, they comprise only 38% of foundational computer science classes. Disparaties like these point to more than an access issue (Code.org 2024; Fletcher & Warner, 2021; Warner et al, 2022).

To address low CS participation, both overall and particularly among learners from underrepresented groups, recent efforts have increasingly focused on introducing CS instruction at lower grade levels (English, 2017; Madrigal et al., 2020; Wiebe et al., 2019; Weintrop et al, 2018). Research suggests that these early interventions are critical because students are still forming their perceptions of gender roles and career trajectories (Barker & Aspray, 2006; Whitecraft & Williams, 2010) and can benefit from early exposure to CS concepts that encourage continued engagement in high school and beyond (Lee et al., 2023; Denner, 2011; Ryoo et al., 2012). Notably, middle school (MS) CS classrooms tend to reflect a more demographically representative student population than HS CS classrooms (Code.org, 2024). This is likely attributable to the fact that MS CS instruction is often integrated into mandatory curricula in the U.S., making it not only accessible to students, but required.

Considering the prior research described above as a whole, a clear message emerges: despite the US' increasing efforts to broaden participation in CSE, data consistently show that participation remains closely tied to sociodemographic factors and CS classrooms often do not represent the diversity of the schools they serve. Many initiatives and projects are working to address this, but systemic changes of this size require sustained efforts and impacts may not be immediate. For instance, changes to K-12 educational policies may take over a decade to realize their impacts and may be difficult to study at earlier points in time. Nevertheless, a deeper understanding of CS pathways overall is crucial for identifying when, how, and why students choose to engage with CS. This paper contributes to the growing body of research by examining the transition from 8th grade to 9th grade as a pivotal decision point in the K-12 CS pipeline.

2. Methods

Through a federally funded grant centered on the enhancement of CS and related instruction at the MS level, eight U.S. public school districts in a northeastern state of the provided MS and HS administrative data. Data was considered for all students in grades 6-12 between 2018 and 2024, and included sociodemographic characteristics for each student, along with their courses, marking period grades, and final course grades for each academic year. Data from the eight districts and six academic years were combined, retaining student, school, and year identifiers. Variables were standardized to account for variations in individual schools' reporting.

Prior to analysis, racial and ethnic categories with small sample sizes (Indigenous, Multiracial, and Pacific Islander) were recoded to an "Other" category. Similarly, students' gender was recoded as "male" or "female and non-binary" to retain students with other gender identities (supported only by some schools' SISs) and to address their small group size. Several key indicator variables were created to capture student characteristics: English Language Learner (ELL) status, receipt of special education services, economically disadvantaged status, enrollment in Algebra 1 (the U.S. elementary algebra course that introduces variables and equations), in middle school, and enrollment in a CS and CS-related course in 9th grade. Two composite variables were created for this analysis: (1) the total number of CS and CS-related classes taken in middle school and (2) the average final course grade across all middle school courses. These variables were designed to provide an understanding of students' academic preparation and course-taking patterns that may impact the critical transition from middle to high school.

The unit of analysis in this study was an individual student enrolled in any participating district with available data for both 8th and 9th grades. A total of 5,505 students met this inclusion criteria, and their characteristics are presented in Table 1. The sample comprises a diverse population of students, with a varied racial composition: 46% Hispanic, 34% White, 11% Black, 7% Asian, and 2% Other. Additionally, the sample includes representation from key subgroups: 18% of students receive Special Education services, 9% are ELLs, and 56% are classified as Economically Disadvantaged. To examine predictors of CS-related course enrollment in 9th grade, we applied a binary logistic regression model using the sociodemographic variables described above.

For the purposes of this study, "CS courses" were defined as a manually verified subset of courses classified under either category 11 (Communication and Audio/Visual Technology) or category 21 (Engineering and Technology) in the School Courses for the Exchange of Data (SCED) classification system (National Forum on Education Statistics, 2014). Additional predictor variables, discussed above, were included based on prior research and theoretical relevance, such as the relationship between Algebra 1 completion and computer science outcomes (Torbey et al., 2020) and the role of pre-high school exposure to computer science in fostering longer-term CS participation (McGee et al., 2017). To facilitate interpretation of effect sizes, odds ratios (ORs) and 95% confidence intervals (CIs) were calculated. All analyses were conducted using R version 4.4.1 (R Core Team, 2024).

Table 1. Sample Characteristics

Characteristic	N = 5,505[1]
Gender	
Male	2,860 (52%)
Female	2,642 (48%)
Other	3 (<0.1%)
Race/Ethnicity	
White	1,860 (34%)
Asian	387 (7.0%)
Black	606 (11%)
Hispanic	2,510 (46%)
Indigenous	2 (<0.1%)
Multiracial	123 (2.2%)
Pacific Islander	17 (0.3%)
Sp. Education	992 (18%)
ELL	488 (8.9%)
Economically Disadvantaged	3,104 (56%)
Avg. MS Course Grade	0.89 (0.83, 0.94)[2]
Algebra I in MS	1,563 (28%)

[1] n (%); [2] median (Q1, Q3)

3. Results

This study examines the predictors of 9th grade CS enrollment using a logistic regression model, shown in Table 2. The model was constructed using a sample of 5,505 9th grade students from eight public school districts in a northeastern state in the U.S., of whom 1,463 (27%) enrolled in a CS course. The model demonstrates a statistically significant improvement over the null model ($\chi^2 = 351.5$, $p < 0.001$), indicating that the included predictors collectively contribute to explaining variability in students' enrollment. Below, we describe the patterns observed across sociodemographic and academic factors.

3.1 Sociodemographic Factors

Gender emerged as one of the strongest predictors of enrollment. Male students were substantially more likely to take a CS course compared to non-male students (log-odds = 0.90, $p < 0.001$). In descriptive terms, 35% of male students enrolled, while fewer than 18% of female and non-binary students did so. This resulted in classrooms where male students outnumbered female and non-binary students by nearly two to one.

Racial and ethnic differences were also evident. Asian students showed significantly higher odds of enrolling in CS compared to White students (log-odds = 0.45, $p < 0.001$), with nearly 40% of Asian students in the sample participating, compared to 26% of White students. Hispanic students also showed higher odds than White students (log-odds = 0.17, $p = 0.044$), with about 26% enrolling. The participation rate for Black students, 22%, did not differ significantly from that of White students (log-odds = 0.05, $p = 0.7$).

Additionally, English Language Learner (ELL) status was negatively associated with CS course enrollment. Only 19% of ELL students participated in 9th grade CS, compared to 27% of their non-ELL peers (log-odds = -0.34, $p = 0.008$). The model also tested for differences by students' economically disadvantaged status, but no significant relationship was found between students' economic status and their likelihood of enrolling in CS (log-odds = 0.08, $p = 0.3$) in this sample.

Table 2. Fitted Logistic Regression Model

Characteristic	log(OR)[1]	95% CI[2]	p
Gender (male)	**0.90**	**0.77, 1.0**	**<0.001**
Race/Ethnicity			
White	—	—	
Asian	**0.45**	**0.21, 0.69**	**<0.001**
Black	0.05	-0.19, 0.28	0.7
Hispanic	**0.17**	**0.00, 0.34**	**0.044**
Other	0.36	-0.05, 0.75	0.076
Sp. Education	**-0.27**	**-0.45, -0.10**	**0.002**
ELL	**-0.34**	**-0.59, -0.09**	**0.008**
Economically Disadvantaged	0.08	-0.07, 0.23	0.3
CS-related Courses Taken by Grade 8	**1.1**	**0.69, 1.5**	**<0.001**
Avg. MS Course Grade	**2.3**	**1.2, 3.3**	**<0.001**
Algebra I in MS	0.19	0.03, 0.35	0.022

[1] OR: Odds Ratio; positive log-odds associated with increases in predicted response; [2] CI: Confidence Interval

3.2 Academic Factors

Overall academic achievement, measured by middle school course grades, was a significant and positive predictor (log-odds = 2.3, $p < 0.001$). Although grade differences were small in absolute terms, the strength of the association suggests that students with higher academic performance overall are more likely to pursue CS in high school, as illustrated in Figure 1.

The strongest academic predictor in this model was prior CS-related course participation in middle school (log-odds = 1.1, $p < 0.001$), meaning that students with prior exposure to CS content were substantially more likely to continue in the CS pathway in high school. Results also suggest that Algebra I plays a role in 9th grade CS enrollment: completion of this course in middle school was a significant predictor (log-odds = 0.19, $p = 0.022$). Among students who completed Algebra I before high school, 33% went on to take CS in 9th grade, compared to 24% of those who took Algebra I in high school.

Students who received special education services at any point were less likely to enroll in a CS course (log-odds = -0.27, $p = 0.002$). While 27% of students without special education services took a CS course, this figure dropped to 23% for students receiving such services.

4. Discussion

This study provides a novel perspective by focusing specifically on 9th grade students, a critical entry point for CSE; the study examined academic and sociodemographic predictors of 9th grade CS course enrollment. In our

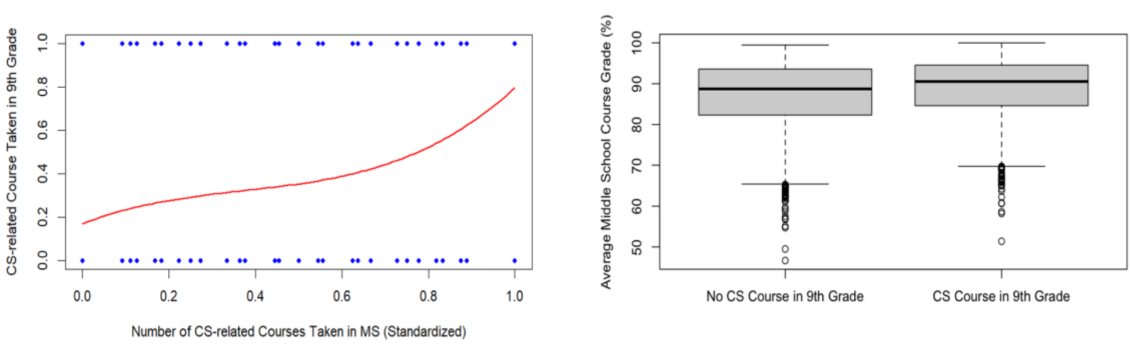

Figure 1. Middle School Academics & 9th Grade CS Participation

sample, 27% of 9th grade students enrolled in a CS course. Recent data from Code.org (2024) shows that 6.4% of high school students (grades 9–12) took a CS course in the 2023–2024 school year. While these percentages describe different populations and cannot be directly compared, their stark difference raises important considerations. Specifically, our results suggest that 9th grade may represent a particularly promising entry point for expanding participation in computing. It is possible that students are more open to trying new subjects like CS at the start of high school, when they are building their academic identities and course patterns (Alhadabi & Li, 2020; Gottfried, 2022). While more longitudinal research is needed to confirm this progression, these findings point to 9th grade as a strategic moment to concentrate outreach and recruitment efforts, both to boost immediate participation and to support sustained engagement over time.

At the same time, our data show that even when courses are available, participation is not evenly distributed. Our model revealed persistent disparities based on gender, race, language status, and academic background. However, it is important to note that our model is predicated on a CS course being available to 9th graders. This mirrors patterns reported in national studies (Code.org, 2024; Warner et al., 2021) and reinforces a critical message from the literature: expanding access to CS courses is necessary, but not sufficient, to achieve broad or equitable participation. Moving toward universal participation requires not only offering courses, but also ensuring that all students feel invited, supported, and capable of succeeding in them.

4.1 Sociodemographic Factors: Interpretation and Implications

Figure 2 illustrates participation in 9th grade CS courses by students' sociodemographic groups. Gender emerged as the strongest sociodemographic predictor in our model, with male students significantly more likely to enroll in 9th grade CS than their female and non-binary peers. This pattern is consistent with long-standing national trends (Code.org, 2024) and existing research (Jaccheri & Fast, 2020). These early imbalances are concerning because they reinforce the perception of CS as a male-dominated space, which can discourage participation and persistence among students who do not see themselves reflected in these classrooms (Venkataraman et al., 2019; Weston et al., 2019). Addressing this imbalance requires not just offering CS courses, but also actively challenging gendered stereotypes about who belongs in CS.

Racial and ethnic patterns in our findings both align with and complicate national trends. National data show that Black, Hispanic, and Native American students are less likely to attend schools that offer CS courses (Code.org, 2024), reflecting systemic disparities in *access*. In our sample, limited to schools where CS was already

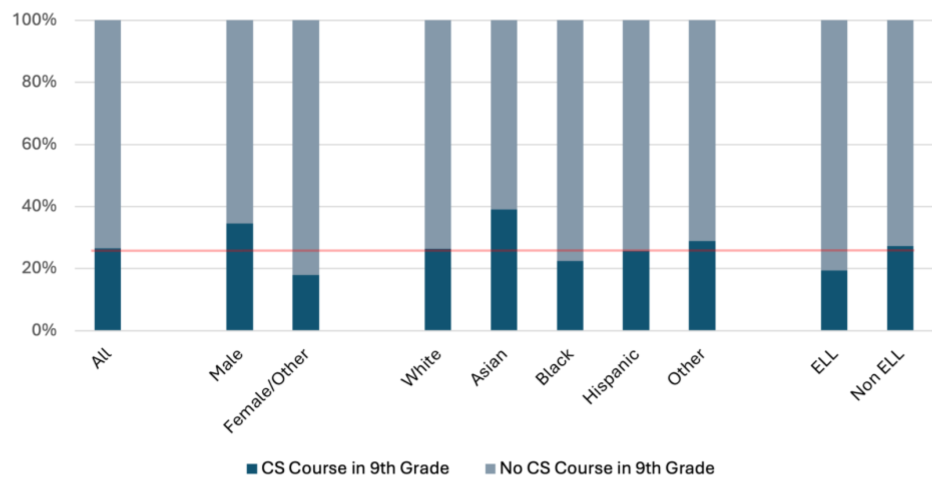

Figure 2. Sociodemographic Characteristics & 9th Grade CS Participation

available, we found that Hispanic students were slightly *more* likely than White students to enroll, and while Black students participated at slightly lower rates, though the difference was not statistically significant. These findings suggest that when CS is made available, participation gaps may narrow for some groups, though not universally. Asian students in our sample were the most likely to enroll, consistent with research (Code.org, 2024). However, we note that aggregating subgroups into a single category risks masking disparities among AAPI communities.

Economic disadvantage was *not* a significant predictor in our model. However, this result should be viewed considering the study's design. As discussed earlier, schools with higher economically disadvantaged student populations are less likely to offer CS, so the present study likely systematically excludes schools (and by extension, students) that serve these communities. Thus, while our model did not find a statistically significant relationship, this does not mean that economic inequality is irrelevant and should not be interpreted as a contradiction or argument against current research. Rather, it once again highlights the difference between *access* and *participation*, both of which must be addressed to achieve broader equity in CSE.

Finally, students identified as English Language Learners (ELLs) and those receiving special education services were significantly less likely to enroll in 9th grade CS. While this could partially reflect language barriers or the need for differentiated instruction, it may also reflect deeper structural and social barriers, such as stigma associated with language learning or special education status (Hagiwara & Rodrigues, 2021; Lei & Allen, 2022). Additionally, scheduling conflicts with required support services may limit students' ability to take elective courses like CS (Blitz et al., 2025).

4.2 Academic Factors: Interpretation and Implications

Figure 3 illustrates participation in 9th grade CS courses with students grouped along vexplored academic factors. Among the academic factors examined, prior CS course-taking in middle school emerged as the strongest predictor of 9th grade enrollment, reinforcing the growing body of research highlighting the value of early exposure (English, 2017; Lee et al., 2023; McGee et al., 2017). This finding strengthens the case for making CS an integrated part of middle school curricula, particularly in formats that reach all students, not just those who opt in.

Completion of Algebra I in middle school also showed a statistically significant positive relationship with 9th grade CS enrollment. This aligns with prior work that links math acceleration to increased engagement in technical subjects (Amiel & Blitz, 2022; Torbey et al., 2020). However, this pattern raises important considerations: unlike math, which is typically structured with differentiated pathways that allow for both advanced and on-grade-level progression, CS is rarely offered in similarly tiered models. This risks creating a pipeline that disproportionately favors students already on accelerated academic tracks. This is further supported by our finding that students with higher overall academic performance were significantly more likely to enroll in CS in 9th grade.

As shown earlier in Figure 1, 9th grade CS enrollment steadily increases with students' average middle school grades, suggesting that students who see themselves as "average" or who struggle academically may self-select out of CS, viewing it as too difficult or specialized. This is consistent with concerns raised in the literature about the positioning of CS as a domain requiring advanced mathematical or analytical aptitude (Amiel & Blitz, 2022; CSTA et al., 2022; NASEM, 2024).

5. Conclusion

This study adds to the conversation on broadening participation in CSE by presenting the transition from middle school to high school (the transition to 9th grade, for U.S. studnets) as a pivotal moment in the CS pipeline. While recent policies have expanded the availability of CS courses, our findings reinforce that access alone is not enough. Participation remains uneven, with significant disparities by gender, language status, special education status, and academic achievement.

The start of high school represents both a reflection of students' prior experiences and a critical opportunity for schools to influence future engagement. In our sample, 27% of 9th graders enrolled in a CS course; although not directly comparable, a interesting figure to consider alongside the national 6.7% (Code.org, 2024) for all HS students. This suggests that 9th grade may be an effective point to invite a wide, diverse student population to CS. However, gaps are already present by this point, shaped by factors such as students' earlier exposure to CS, their academic histories, and their sense of who "belongs" in CS.

Our results show that academic background strongly predicts 9th grade enrollment. Students who took CS in middle school, completed Algebra I early, and achieved higher middle school grades were more likely to enroll in CS in 9th grade. This raises concerns about how CS is presented in school systems; often positioned as advanced or math-intensive, accessible primarily to students already on accelerated tracks. This framing risks turning academic performance into an unintended gatekeeper.

Because of this, the 8th-to-9th grade transition stands out as a practical checkpoint both for researchers and for educators. For researchers, it provides a timely opportunity to study how early exposure and systemic factors shape students' course-taking decisions. For educators and school systems, it is a strategic moment to intervene. Schools that limit CS to upper grades should consider opening enrollment earlier, adjusting schedules, and reducing unnecessary barriers. Outreach efforts must begin before course selection, reaching all students and clearly communicating that CS is for everyone, not just those with advanced math skills or prior coding experience.

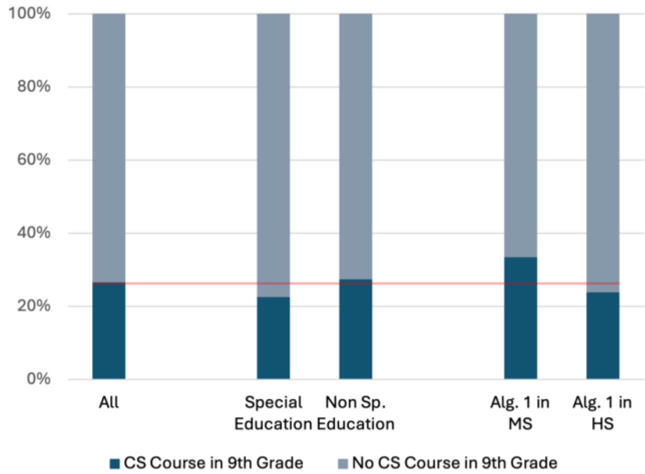

Figure 3. Academic Predictors & 9th Grade CS

At the same time, schools must reflect on how internal structures and resource limitations might unintentionally undermine their goals. While not all schools can offer specialized sections or fully differentiated instruction, they can work to ensure that scheduling, prerequisites, and messaging do not systematically exclude students who require additional supports. Addressing misconceptions about the difficulty or exclusivity of CS is essential, not only to increase participation but to advance the broader goal of more universal participation in CSE, which provides a critical skillset for today's, and tomorrow's, society and workforce.

5.1 Limitations & Future Work

This study focuses on schools where 9th grade CS courses were already available, excluding students in schools without such offerings from the model; these students are more likely to be from populations typically underrepresented in CSE. As such, results should be interpreted through this lens, drawing a clear distinction between predicting access and predicting participation, as presented in the discussion. Additionally, while our dataset included a large and diverse student population, smaller subgroup sizes for certain characteristics, such as ELL and special education status, may limit the precision of some subgroup-specific findings.

Future research should build on this work by further investigating 9th grade as a pivotal transition point, including the factors that influence course selection when CS is available. Mixed-methods studies could deepen understanding of how students make enrollment decisions, including the roles of prior experience, perceptions of course difficulty, and school-level messaging. Larger or multi-site studies could also strengthen subgroup analyses, particularly for multilingual learners and students receiving special education. We also note that the analysis conducted in this study, a binary logistic regression, does not yield causal findings, and controlled experiments that can generate causal insight may offer a promising area for future work. Additional research of any kind could contrinbute to a more nuanced understanding of the interplay between access and participation, identifying how expanding course availability interacts with student-level factors to shape engagement in CS over time.

Acknowledgments

This research was conducted with support from the U.S. Department of Education Office of Elementary & Secondary Instruction under Educational Innovation & Research (EIR) Award #S411C200084.

References

Adrion, W. R., Dunton, S. T., Ericson, B., Fall, R., Fletcher, C., & Guzdial, M. (2020). U.S. states must broaden participation while expanding access to computer science education. *Commun. ACM*, 63(12), 22–25. https://doi.org/10.1145/3430375

Alhadabi, A., & Li, J. (2020). Trajectories of Academic Achievement in High Schools: Growth Mixture Model. *Journal of Educational Issues*, *6*(1), 140. https://doi.org/10.5296/jei.v6i1.16775

Amiel, D., & Blitz, C. (2022). Computer science teacher capacity: The need for expanded understanding. *International Journal of Computer Science Education in Schools*, *5*(4). doi.org/10.21585/ijcses.v5i4.151

Barker, L. J., & Aspray, W. (2006). The state of research on girls and IT. In J. M. Cohoon and W. Aspray, *Women and Information Technology: Research on Underrepresentation*, pp. 3-54. Cambridge, MA: MIT Press.

Barr, V. (2022). What must all post-secondary students learn about computing? *Commun. ACM*, *65*(11), 29–31. https://doi.org/10.1145/3563968

Blitz, C.L., Duncan, T., & Amiel, D.J. (2025). *The Intersection of computer science education and ELL services: Understanding the impact on underrepresented learners.* Unpublished manuscript.

Bureau of Labor Statistics, U.S. (2024). *Occupational outlook handbook: Computer and information technology occupations*. Retrieved from https://www.bls.gov/ooh/computer-and-information-technology/

Code.org (2024). *2024 state of computer science education.* Retrieved from https://advocacy.code.org/stateofcs

CSTA, IACE, ACM, Code.org, College Board, CSforALL, & ECEP Alliance (2024). *Reimagining CS pathways: Every student prepared for a world powered by computing*. Association for Computing Machinery. https://doi.org/10.1145/3678016

Denner, J. (2011). What predicts middle school girls' interest in computing? *International Journal of Gender, Science and Technology*, *3*(1), 53-69.

English, L. (2017). Advancing elementary and middle school STEM education. *International Journal of Science and Mathematics Education*, *15*(S1), 5–24. https://doi.org/10.1007/s10763-017-9802-x

Fletcher, C.L., Warner, J.R. (2021, February). CAPE: A framework for Assessing equity throughout the computer science education ecosystem. *Communications of the ACM*, 64(2), 23-25.

Hagiwara, S. & Rodriguez, N. J. (2021). English learners (EL) and computer science (CS) learning: Equity issues. In J. Keengwe & Y. Tran (Eds.), *Handbook of Research on Equity in Computer Science in P-16 Education* (pp. 70-87). IGI Global Scientific Publishing. https://doi.org/10.4018/978-1-7998-4739-7.ch005

Gottfried, M. (2022). Linking applied STEM coursetaking beyond STEM outcomes: Examining high school completion and college-going behavior. *The High School Journal*, *105*(3), 175–198.

Jaccheri, J, Pereira, C, & Fast, S. (2020). Gender Issues in Computer Science: Lessons Learnt and Reflections for the Future. *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 9–16. https://doi.org/10.1109/SYNASC51798.2020.00014

Lee, H., Closser, F., Alghamdi, K. *et al*. (2023). Current experiences and factors of future enrollment in computer science for high school students. *TechTrends* 67, 955–964. https://doi.org/10.1007/s11528-023-00906-1

Lei, Y., & Allen, M. (2022). English language learners in computer science education: A scoping review. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*, 57–63. https://doi.org/10.1145/3478431.3499299

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational Thinking Is More about Thinking than Computing. *Journal for STEM Education Research*, *3*(1), 1–18. https://doi.org/10.1007/s41979-020-00030-2

Madrigal, V., Yamaguchi, R., Hall, A., and Burge, J. (2020). *Promoting and supporting computer science among middle school girls of color: Initial findings from BRIGHT-CS*. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 247–253. https://doi.org/10.1145/3328778.3366855

Margolis, J., Ryoo, J. J., Sandoval, C. D. M., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, *3*(4), 72–78. https://doi.org/10.1145/2381083.2381102

McGee, S., McGee-Tekula, R., Duck, J., Greenberg, R. I., Dettori, L., Reed, D. F., Wilkerson, B., Yanek, D., Rasmussen, A. M., & Chapman, G. (2017). Does a taste of computing increase computer science enrollment? *Computing in Science & Engineering*, *19*(3), 8–18. https://doi.org/10.1109/MCSE.2017.50

Michaeli, T., Seegerer, S., & Romeike, R. (2023). *What Students Can Learn About Artificial Intelligence – Recommendations for K-12 Computing Education*. https://arxiv.org/abs/2305.06450

National Academies of Sciences, Engineering, and Medicine (NASEM; 2024). *Equity in K-12 STEM education: Framing decisions for the future*. Washington, DC: The National Academies Press. https://doi.org/10.17226/26859

National Center for Education Statistics (NCES; 2019). *Percentage of persons' ages 18-29 enrolled in colleges or universities, by age group, 4-category local, and sex*. Retrieved from nces.ed.gov/pubs2015/2015144.pdf

National Forum on Education Statistics. (2014). *Forum guide to school courses for the exchange of data (SCED) classification system. (NFES 2014-802).* U.S. Department of Education. Washington, DC: National Center for Education Statistics. https://nces.ed.gov/forum/pub_2014802.asp

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Ryoo, J.J., Margolis, J., Lee, C.H., Sandoval, C.D. M., & Goode J. (2012). Democratizing computer science knowledge: Transforming the face of computer science through public high school education. *Learning, Media and Technology*, 38(2), 161-181.

TeachAI and CSTA (2024). *Guidance on the future of computer science education in an age of AI*. Retrieved from https://www.teachai.org/cs

Torbey, R., Martin, N. D., Warner, J. R., & Fletcher, C. L. (2020). Algebra I Before High School as a Gatekeeper to Computer Science Participation. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 839–844. https://doi.org/10.1145/3328778.3366877

Venkataraman, R., Agarwal, E., & Brown, D. W. (2019). Engaging K-12 Students Essential for Reducing Gender Gap in Computer Science Education. In G. H. Marks (Ed.), *Proceedings of International Journal on E-Learning 2019* (pp. 331–343). Association for the Advancement of Computing in Education (AACE). https://www.learntechlib.org/p/149070

Warner, J. R., Fletcher, C. L., Martin, N. D., Baker, S. N., (2022). Applying the CAPE framework to measure equity and inform policy in computer science education. *Policy Futures in Education*, 0(0), 1-19. https://doi.org/10.1177/14782103221074467

Weston, T. J., Dubow, W. M., & Kaminsky, A. (2019). Predicting Women's Persistence in Computer Science- and Technology-Related Majors from High School to College. *ACM Trans. Comput. Educ.*, *20*(1). https://doi.org/10.1145/3343195

Wiebe, E., Barnes, T., Freeman, S., Frye, D., Maher, M. L., Cao, L., Dorodchi, M., Pugalee, D., Rorrer, A., Boulden, D., & Catet́é, V. (2019, February). *Developing a systemic, scalable model to broaden participation in middle school computer science*. In 2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) (pp. 1-2). IEEE.

Weintrop, D, Hansen, A.K., Harlow, D.B., Franklin, D. (2018). *Starting from scratch: Outcomes of early computer science learning experiences and implications for what comes next*. In Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18). Association for Computing Machinery, New York, NY, USA, 142–150. https://doi.org/10.1145/3230977.3230988

Whitecraft, M.A., & Williams, W.M. (2010). Why aren't more women in computer science? In *Making Software: What Really Works and Why We Believe It*, ed. A. Oram and G. Wilson, 221-238. Sebastopol, CA: O'Reilly Media, Inc.

Copyrights

Designing a Student-Centric Computer Science Curriculum: Enabling Flexibility and Personalised Learning in Secondary Schools

Ajay Kumar Yadav[1]

Dil Prasad Shrestha, PhD[2]

[1]Project Director, Management Innovation, Training and Research Academy Pvt. Ltd., Kathmandu Nepal

[2]Executive Director, Management Innovation, Training and Research Academy Pvt. Ltd., Kathmandu Nepal

DOI: 10.21585/ijcses.v7i2.235

Abstract:

This study analyses the Computer Science (CS) curricula for grades 9 and 10 in Nepal, emphasising students' interests and needs within the social context. The study applied the mixed-methods research design. The quantitative data were collected from questionnaire surveys with students and teachers. The qualitative information was collected from curriculum designers, textbook authors, school principals, teachers and students from the Kathmandu valley, Nepal. Random and purposive sampling techniques were used to collect data from the respondents and participants. Data and information were analysed using both quantitative and thematic techniques. The study revealed significant opportunities to enhance the CS curricula by integrating interdisciplinary concepts supported by pragmatic examples. It also highlighted that improving the CS curricula for grades 9 and 10 is a top priority for both schools and students. The study proposes a student-centric CS curriculum development framework that balances between foundational concepts of CS and students' interests. This framework would address the diverse needs of students, considering their physical and mental abilities and interests in their learning. It also suggests that CS students should be included in the curriculum development committee so that they can provide practical feedback and suggestions in the CS curriculum revision process.

Keywords: student-centred learning, education policy, computer science curriculum

1. Introduction

The secondary school-level curriculum is a formal education plan that includes syllabi, textbooks, theory and practical teaching hours for an academic session. The curriculum serves as a reflection of the nation's future by equipping students with the essential skills and knowledge. Curriculum development involves several educational stakeholders, often resulting in substantial cost, effort and time.

Computer Science (CS) is "the systematic study of algorithmic processes that describe and transform information: theory, analysis, design, efficiency, implementation, and application" (Comer et al., 1989, p. 12). CS is an interdisciplinary subject that incorporates knowledge and applications from several domains and disciplines. Rapid innovations and transformations in CS have continuously reshaped industries, human lifestyles, and the education system (Matthew & Okafor, 2021; Webb et al., 2017). In the coming years, several jobs are predicted to be replaced with advancements in smart technology, artificial Intelligence, robotics, and algorithms (Brougham & Haar, 2018).

The Government of Nepal (GoN) statutory body, the Curriculum Development Centre (CDC), is responsible for creating the CS curriculum and textbook for grades 9 and 10 in Nepal (Gharti, Upreti, & Thapa, 2020a, 2020b). CS is an elective subject in the combined syllabus for grades 9 and 10 in Nepal. Students can enrol in the CS subject at grade 9 without any prior study of the CS subject (CDC, 2019a). Grade 10 is considered an "iron gate" for the Nepalese students in their academic career since achieving good marks/grades at grade 10 will impact their future career. Therefore, guardians (parents) put their best effort in consideration of time and expenses for their children's education. However, there is considerable dissatisfaction among students with the knowledge and skills learned from the CS course.

Nepal has a multi-linguistic and multi-cultural society, thus students from different native languages and cultures are enrolled in the school. Each student is different with their own personal preference in learning aligned with their culture and language. Saloviita (2020) emphasises that teachers' attitudes are crucial for an inclusive education in the classroom that prioritises individual student needs. The previous studies were mostly focused on the majority of student populations and their learning style. Thus, there is a need to identify individual student preferences with their challenges in diverse learning environments (Zhansulu et al., 2022). While there's growing emphasis on student-centred learning, there remains a gap in providing students with the essential science and technology knowledge that is applicable in social contexts (Israel et al., 2020; Neupane, 2020; Stetter, 2018). Therefore, this study analyses the scope of personalised learning in the existing CS curriculum and the education policy for secondary schools in Nepal.

1.1 Objectives

The study aims to analyse the extent to which the existing CS Curriculum is designed in terms of students' personalised and flexible learning for secondary schools, grades IX and X in Nepal. It also aims to propose a student-centric CS Curriculum development framework that aligns with students' interests and abilities in the given social context.

2. Literature Review

2.1 Theoretical Foundations of Learning

Learning theories provide an understanding of the learning process in students. Behavioural theory emphasises structured learning environments with specific objectives and reinforces desired behaviours and actions through learning outcomes by providing reward and punishment techniques. Cognitive learning theory emphasises individuals' perception, memory, and problem-solving to understand the learners' knowledge construction process based on their age and abilities (Bonk & Cunningham, 2012). Connectivism emphasises learning as a distributed process from online resources and networking (Siemens, 2005).

According to William Pinar (2019), curriculum is a lived educational experience in the learner's personal, social, and cultural contexts rather than a fixed set of learning objectives and outcomes. He emphasises curriculum design from multiple stakeholders' perspectives in the social context. Students are required to actively participate and engage in the learning process by making personal connections with learning plans and experiences.

2.2 Frameworks for Curriculum Design and CS Curriculum

Tyler (1949) introduced a linear model with four components: objectives, activities, organisation, and evaluation. Taba (1962) expanded the Tyler four-step model into a seven-step linear model by giving more emphasis on the educational purposes for curriculum development. Wheeler (1967) proposed a cyclic model by incorporating a feedback loop evaluation as an input required for the curriculum revision process (Kelly, 2004). The 21st-century skill-sets emphasise three key areas: learning skills (critical thinking, creativity, communication, collaboration), literacy skills (information literacy, media literacy, technology literacy), and life skills (flexibility, leadership, initiative, productivity, self-awareness) (CBSE, 2023a; Nouri et al., 2019; Van Laar et al., 2020).

The Computer Science (CS) curricula of the CBSE, India and Bhutan emphasise application-based learning, incorporating Python programming languages to foster practical skills (CBSE, 2023b; MoEB, 2021). Pakistan's CS curriculum prioritises fundamental programming concepts through C/C++, reflecting a more traditional approach (MoEP, 2009). However, all three curricula exhibit a significant gap in 'unplugged' programming—a pedagogical approach that teaches computational thinking without electronic devices (Bell et al., 2009). This omission is notable, given global trends toward inclusive and low-resource CS education strategies (CS Unplugged, 2022).

Wilson (2011) conceptually discusses student-centric customisation in three dimensions—flexibility in content selection, scheduling, and completion duration. Powers (2003) emphasises a "breadth-first" model to incorporate interdisciplinary topics for introductory programming courses in CS.

Readability is a criterion for selecting a textbook with regard to the content and arrangement of content. Flesch Reading Ease score and Flesch-Kincaid Grade Level are readability tools used to judge the language of a textbook in terms of word, sentence and paragraph length as per grade standard (Flesch, 1948; Zhou et. al., 2017). The K–12 CS Framework (2016) and CBSE (2023b) CS curriculum emphasise core concepts, practical skills, and equitable access to prepare students for the digital era. Additionally, Bloom's Taxonomy provides a structured approach to designing educational plans, enabling teachers to address diverse cognitive levels and personalise instruction (Anderson & Krathwohl, 2001).

Computational thinking involves breaking down problems into smaller sub-problems, recognising patterns, selecting appropriate algorithms, and developing solutions (Ezeamuzie & Leung, 2022; Yadav & Stephenson, 2016). Students can initially learn computational thinking through unplugged programming activities before implementing solutions in a programming language (PL). The program gives instructions to computers to perform specific tasks. Block-based programming serves as a scaffolding tool, helping learners understand CS concepts without the complexity of PL syntax (Weintrop & Wilensky, 2019). Text-based programming requires prior knowledge of PL syntax and semantics. Accessibility tools are required to meet the diverse needs of minority students (Salas-Pilco et al., 2022).

2.3 Teaching and Learning Pedagogy

TSPACK[1] and TPACK[2] models emphasise pedagogy and content in a social context of knowledge through collaborative learning (Arce-Trigatti et al., 2019; Mishra & Koehler, 2006). These models highlight the interplay among pedagogy, content, and technology, specifically incorporating social-context collaboration in the curriculum.

William (2019) mentioned that game-based learning is interactive and enhances creativity, cooperation, critical thinking, and problem-solving skills in the classroom. Project-based learning includes interdisciplinary knowledge and activities. Students are engaged in learning, discussion, and cooperation among team members in verbal and non-verbal modes in project-based learning (Hanna, 2008). Game-based learning creates a stimulating educational environment with emotions reflecting those compared to textbook materials (Kuzu & Durna, 2020).

2.4 Educational Policies in Nepal

Article 31 of the Constitution of Nepal highlights the fundamental rights related to education, inclusivity, and socio-economic areas. Article 32 mentions preserving and promoting own language, culture, and scripts. Article 39 mentions having children's development and participation programs as per their ability (MoLJPA, 2015). The National Curriculum Framework (NCF) underscores the importance of the local need-based curriculum aligned with the social context and teacher development programs for the effective delivery of the school-level curriculum (CDC, 2019b). While designing a curriculum based on local needs, a Local Curriculum Development Committee (comprising educationists, historians, teachers, guardians, people with disability, representatives of the people of different ethnic groups and communities) should be formed. Additionally, the NCF emphasises that the curriculum will be developed based on the child-centred approach (CDC, 2019b).

The Education Policy in Nepal has emphasised digital literacy at the basic education level[3] and secondary education level[4] (MoEST, 2019). The current CS curriculum has equal weightage (50%) for both theory and practical assessments (CDC, 2019a). The policy further recommends the locally-based curriculum; however, the same textbook is recommended nationwide to be taught in the secondary schools. The assessment has different patterns of questions set by provinces, but the same textbook and content are taught in the schools.

3. Methodology

This study employed a mixed-methods design, collecting and analysing both quantitative and qualitative data to comprehensively understand the CS curriculum in Nepal. The questionnaire survey tools consisted of three sections: general information of the respondents, opinions of CS teachers and CS students, and improvement of the CS curriculum. The qualitative tools incorporated information related to the CS curriculum, national curriculum framework, educational policy, and contemporary technology.

The randomly selected CS students (n=418) participated in the survey from 17 schools within the Kathmandu valley, comprising three districts: Kathmandu, Lalitpur, and Bhaktapur. The CS teachers (n=40) were selected purposely from 19 schools in the Kathmandu valley to participate in the study. The Cronbach's alpha value (to

[1]TSPACK: Technological Pedagogical Content Knowledge in a Social Context (Arce-Trigatti et al., 2019) – An extension of TPACK that emphasizes social context knowledge.
[2]TPACK: Technological Pedagogical Content Knowledge (Mishra & Koehler, 2006) – A framework that integrates technology, pedagogy, and content knowledge for effective teaching.
[3] Basic education level in Nepal includes Grades 1 to 8.
4 Secondary education level in Nepal includes Grades 9 to 12.

check the internal consistency of the study tools) for the survey questionnaire was 0.971 for the teachers and 0.936 for the students. According to George & Mallery (2016), these values indicated high internal consistency and reliability of the study tools. Similarly, the content validity was assessed by experts and subject-related professors to check the accuracy of the tool.

4. Data Analysis

The quantitative data were collected from secondary school students (grades 9-12). The demographic profile of the respondents is presented in Table 1.

Table 1. Demographic Profile of the Respondents

Students	Male (n)	Female (n)	Male (%)	Female (%	Total (n)	Total (%)
Students (Grades 9-10)	151	127	58.3	79.9	278	66.5
Students (Grades 11-12)	108	32	41.7	20.1	140	33.5
Total Students	259	159	100	100	418	100
Teaching Experience						
Early Career (< 3 Years)	4	4	15.4	28.6	8	20.0
Intermediate (4-9 Years)	4	3	15.4	21.4	7	17.5
Senior (10 -15 Years)	8	3	30.8	21.4	11	27.5
Expert (16+ Years)	10	4	38.5	28.6	14	35.0
Total	26	14	100	100	40	100
Teachers' Education Qualification						
Bachelor's	10	4	38.5	28.6	14	35
Master's +	16	10	61.5	71.4	26	65
Total	26	14	100	100	40	100

Similarly, qualitative information related to perspectives and improvement of the CS curriculum was collected from the key participants- CS teachers, CS textbook authors, school principals, curriculum designers, and CS students (Grades 9-12) as presented in Table 2.

Table 2. Participants' Information

Participants	Number	Gender	Data Collection Method
CS Teachers	7	Male (5), Female (2)	Focused Group Discussion
CS Textbook Authors	3	Male (3)	Interview
School Principals	2	Male (1), Female (1)	Interview
Curriculum Designers	2	Male (2)	Interview
CS Students (Grades 9-12)	24 (6 students per grade)	Male (3), Female (3)	Focused Group Discussion

Qualitative and quantitative data collected from various sources were analysed based on the following themes and sub-themes (Table 3). According to Braun and Clarke (2006), the thematic analysis technique was applied to analyse the information collected from the participants. The information was categorised into themes, sub-themes, and topics.

Table 3. Themes and Sub-themes

Main Themes	Sub-themes	Topics
Flexible Learning	Curriculum Design	Scope of the Curriculum
		Teaching Hours
		Interdisciplinary Subjects
Personalized Learning	Textbooks	Readability & Contents
		Digital / Print
	Pedagogy	Traditional Lecture
		Lab-Based Learning
		Real-World Applicable (Project/Game)
	Knowledge and Skills	Contemporary Technology
		Technical Skills for Social Context
Education Policy	Flexible Learning	
	Personalized Learning	

5. Results

5.1 Flexible Learning

5.1.1 CS Curriculum

The existing CS curriculum is the fixed plan for all secondary-level students in Nepal. It has a total of 128 teaching hours with equal weightage for theoretical and practical sessions (64 hours each). It has included five programming languages- QBasic, C, HTML, CSS, and database SQL.

Curriculum designers from the CDC mentioned that there is a committee of experts responsible for deciding CS topics; however, not all concerned experts are invited to planning meetings. This limits the inclusion of diverse perspectives and expertise from the professional expert members in the committee.

Textbook authors from the expert committee are responsible for transforming the curriculum plan into a textbook. They also mentioned that CS topics are decided by the CDC. They are not provided with detailed guidelines for incorporating interdisciplinary subjects and social context, due to which they only provide examples in the textbook based on their experiences. The lack of comprehensive guidelines for writing textbooks limits them to include examples related to the social context, further limiting the students ' ability to connect with real-world experiences and broader educational contexts.

CS teachers and CS students were surveyed on the extent to which interdisciplinary subjects, such as Science, Mathematics, Arts and Social Science, are integrated into the CS Curricula for grades 9 and 10 in Nepal. Table 4 presents their responses.

Table 4. Integration of Interdisciplinary Subjects into CS Curricula

Integration of Interdisciplinary Subjects	Teacher	Student
Extensively Integrated	42%	61%
Moderately Integrated	45%	4%
Minimally Integrated	13%	35%

A majority of students (61%) agreed upon the integration of interdisciplinary subjects in the CS curriculum, compared to 42% of teachers, suggesting that teachers could have incorporated examples from other subjects based on their experience. While 45% of teachers and 4% of students expressed moderate views, 13% of teachers and 35% of students mentioned that there is minimal integration of interdisciplinary subjects into the CS curriculum. This showed that there are opportunities for improving the CS curriculum by integrating interdisciplinary concepts with social examples. CS students also mentioned that they have other different interests such as painting, writing essays, poetry, mathematics, scientific experiments, dancing, and singing. According to them, the current CS curriculum does not provide opportunities for exploring these interests through CS applications. The curriculum only focuses on programming languages and theoretical concepts. The grade 10 CS curriculum largely lacks graphics content, limiting students' exposure to creativity and practical applications of CS. Moreover, school principals shared their experiences to include a flexible and customizable CS curriculum that can promote inclusive education for a diverse range of schools and students.

5.2 Personalised Learning

5.2.1 Textbooks

Students mentioned that the present CS textbook has limited learning resources. They are essential for examinations, but to study in detail, they need to refer to several e-learning tutorials. They require teacher guidance even to execute programs from the textbook. The digital version of the textbook has the same materials as the print edition and lacks interactive features. Similarly, CS teachers mentioned that in this digital age, students study more from e-learning platforms due to the availability of abundant examples and interactive video sessions. Textbook authors also highlighted the need for redesigning textbooks, suggesting the inclusion of interactive digital materials accessible via QR codes or URLs printed in the textbook. They also emphasised that a comprehensive CDC curriculum planning process should receive serious attention to develop student-friendly textbooks and sufficient digital learning resources so that students can directly execute programs from digital textbooks. Figure 1 illustrates the results of the survey assessing CS students' confidence in programming concepts.

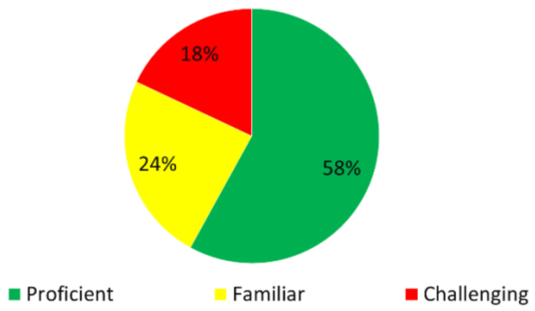

Figure 1. CS Students' Confidence in Programming Concepts

The data reveal that 58% of the students seemed to be proficient in programming, 24% are familiar, and 18% find programming a challenging task. These findings suggest a need for improved teaching-learning strategies in programming. During discussions, students mentioned that programming requires more practice time with different practical examples to understand CS concepts.

CS teachers highlighted that the CS curriculum includes five programming languages (PL) in a limited time of the academic year. They suggested that having only a few PL in an academic session will provide more sufficient time to practice and understand the CS concept. They also noted that the present curriculum emphasises QBasic, which largely does not align with the contemporary PL of the present industry. Additionally, they recommended incorporating block-based PL, as it helps students better understand CS concepts. This showed that the current CS curriculum is content-oriented with several PLs rather than practical, real-world applications.

5.2.2 Pedagogy

CS students were asked about the teaching-learning strategy/pedagogy used in CS Classes. Figure 2 exhibits their responses.

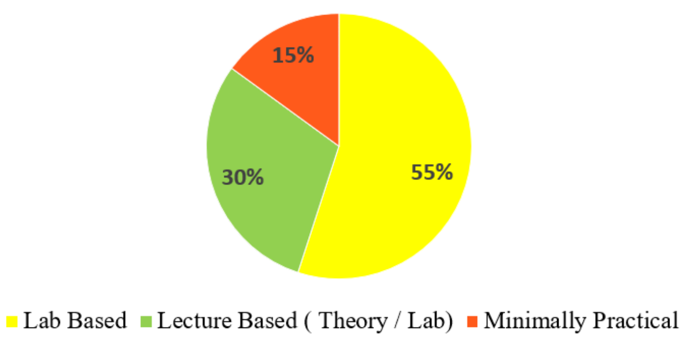

Figure 2. Teaching-Learning Pedagogy Used in CS Classes

The results revealed that 55% of students reported that CS classes are largely Lab-based practical teaching, while 30% mentioned that theory and lab classes are moderately balanced, and 15% indicated that the practical teaching-learning is minimally used. This suggests a need for improvement in teaching strategies to enhance hands-on learning. Students also suggested conducting CS classes in CS laboratories so that they can simultaneously practice programming with CS theoretical classes. This highlights a gap in learning outcomes required in a specific social context.

In the discussion, teachers were asked about their preferred teaching styles in CS classes. Table 5 displays their responses.

Table 5. Teachers' Views on CS Teaching Pedagogy

Teaching Pedagogy	Multi-choice Responses (%)
Functional Programming (Project / Game-based)	87
Practical Skills	77
Simulated/Mock/Industry Exposure Projects	52

The data in Table 2 highlighted that 87% of teachers emphasised functional programming with project implementation or small game development, which enables students to implement CS concepts using a preferred PL to develop applications. Similarly, 77% of teachers highlighted the importance of practical skills, such as language typing, content creation, and graphics. About 52% of teachers emphasised project work similar to real-world industry work. They also mentioned that teaching CS at the foundational level requires engaging examples to captivate interest among students. Since students have studied several subjects till grade 8, they can learn to implement knowledge and concepts from other subjects in the CS programming examples. This demands more focus on project-based curriculum design and pedagogy.

5.3 Curriculum Enhancement

5.3.1 Anticipated Skills and Knowledge Outcomes from the Grade 10 CS Curriculum

CS teachers were consulted regarding the expected skills and knowledge from the grade 10 CS curriculum. They identified some expected competencies of students after completing Grade 10, as shown in Figure 3 (in multiple-choice questions).

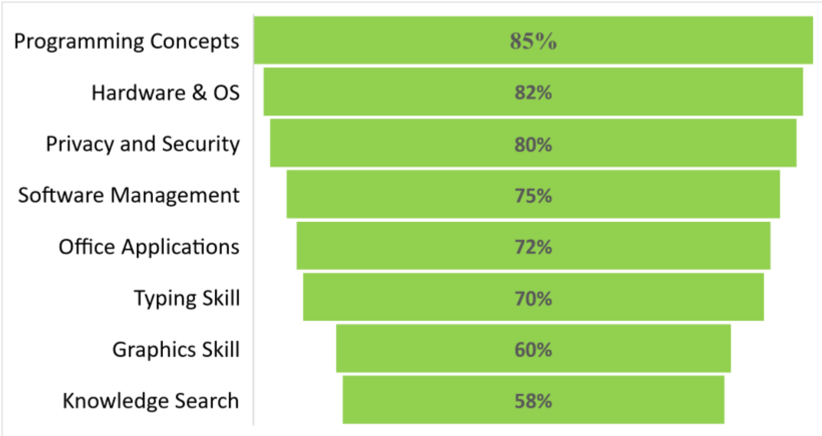

Figure 3. Expected Skills of CS Grade 10 Students

The CS teachers emphasised that students first need to understand programming concepts and develop a simple application and profile website for digital presence. The operational knowledge, related to Operating System (OS), software installation/uninstallation process, and computer hardware, is required to operate the system. Due to social media, students are required to understand the criticality, privacy and security of information for personal data. They are also required to be familiar with office and cloud applications such as Documents, Excel, and PowerPoint. Nepali and English language typing skills are required to present their content. The graphic skills, such as creating images, videos, and animations, are required for an interactive presentation. Moreover, students are likely to have search skills to find relevant information from the web and utilise it in their project work.

The findings from the teachers were discussed with curriculum designers. They mentioned that topics are included in the CS curriculum based on the committee's judgment. Since CS is an elective subject, office applications are included in the compulsory science curriculum at grade 8. However, many science teachers have expressed dissatisfaction with the inclusion of Information and Communication Technology (ICT) skills in the Science curriculum. This highlights the need for a scientific method to determine curriculum topics, rather than relying solely on the committee's judgment. In discussions with CS textbook authors, they emphasised the need to include office applications and typing skills in the CS curriculum.

5.3.2 Technology and Skill for the Revised CS Curriculum

Teachers and students were asked about their views on the enhancement of the revised CS curriculum. The responses from students and teachers revealed a strong priority for integrating technologies and engaging learning approaches, as highlighted in Figure 4 (with multiple-choice questions).

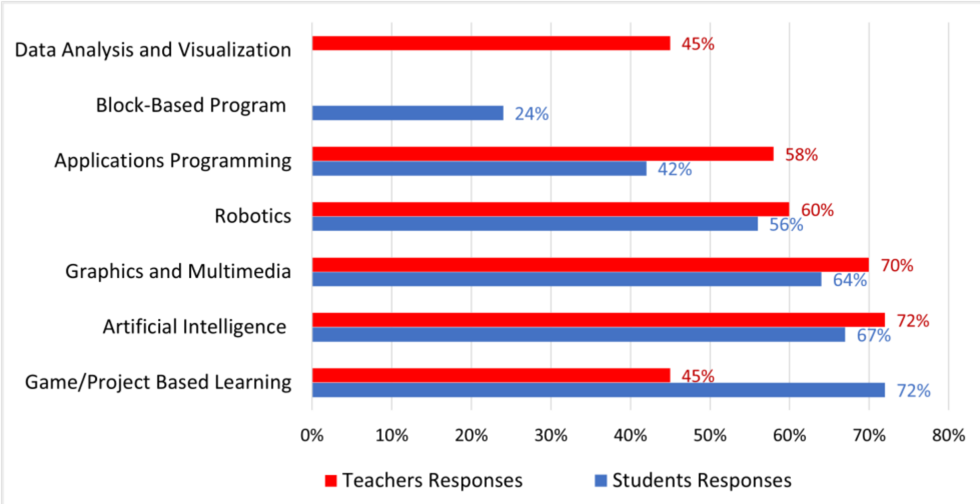

Figure 4. Students' and Teachers' Views on the Revised CS Curriculum

The above bar chart shows that there is a correlation between students' responses and teachers' responses. Students and teachers strongly suggested AI, graphics and multimedia, and Robotics to be included in the CS curriculum. A majority of students (72%) and some teachers (45%) emphasised the inclusion of game/project-based learning. Some teachers (58%) and students (42%) expressed their views to include applications programming in the CS curriculum. While 24% of students emphasised the inclusion of block-based programs, 45% of teachers suggested including data analysis and visualisation in the secondary school-level CS curriculum.

6. Discussion

The study revealed that the CS curriculum and textbooks follow a rigid standardised plan for all schools and students nationwide. Nepal has diversity in terms of geography, having plain land, mountains, and Himalaya's. Moreover, the urban/ rural areas have a vast difference in people's lifestyle and culture. Therefore, the one-size curriculum and textbook often lack to provide holistic CS educational plans. This also emphasises students to learn typing skills in their local languages, such as Nepali, Maithili, Newari, and so on.

The study found that students have various interests in their educational pursuits with their different abilities. Therefore, the CS curriculum plan requires a personalised learning approach with students' interests and abilities to explore them with a wide range of tools available in CS. For example, students interested in painting can explore digital painting. Similarly, students with an interest in Mathematics and analytical thinking could explore programming, data analysis and visualisation. This finding aligns with Zhansulu et al. (2022), emphasising that effective curricula cater to individual learners in a social context.

The study reported that CS curriculum development is a rigorous process due to changes in technologies and requires several stakeholders. This finding aligns with Soto (2015), who states that the curriculum development is a complex and time-consuming process, as curriculum developers have to consider several aspects- social context, students, and subjects- while developing the CS curriculum. The curriculum requires frequent revision due to the changing nature of the CS industry, technology and future workforce, so that students can apply the classroom knowledge and skills in real-world application development. The study reported that the pedagogy is less oriented toward education for real-world application development and contemporary knowledge. Although the National Curriculum Framework (NCF) emphasises life skills and a job-oriented curriculum integrated with

ICT education (CDC, 2019b), the current curriculum, to some extent, still lacks contemporary technologies and programming languages.

In this context, the study suggests including key components in the CS curriculum that provide personalised and flexible learning.

7. Student-Centric CS Curriculum Development Framework

The proposed framework (Figure 6), to be used for designing a student-centric CS curriculum, employs a circular design to organise key components of the CS curriculum. The arrows represent key components such as Program, Graphics, Content Creation (CC), Artificial Intelligence (AI), and Application-Based Learning (ABL).

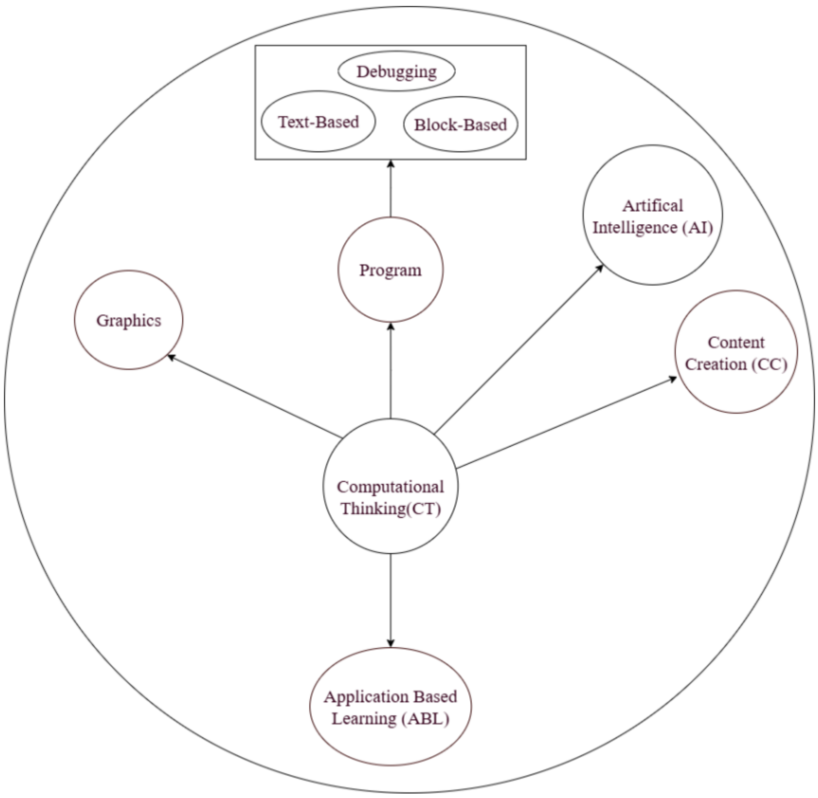

Figure 6. Student-Centric CS Curriculum Development Framework

The key components for designing a CS curriculum include:

Computational Thinking (CT): Computational thinking is learned through both plugged and unplugged methods/techniques. The plugged method uses a computer to execute a program by considering time and space for computation. The unplugged method is used to understand the algorithm and computational thinking without requiring a physical computer device.

Program: The programming is learnt from both approaches of block-based and text-based coding.

Block-Based: This approach of programming is suitable for beginners to understand the semantics of the programming language (PL). This excludes the syntax understanding of PL.

Text-Based: The text-based programming includes both syntax and semantics of PL.

Debugging: This checks that a program performs its intended task by identifying any errors in the code.

Graphics: Students use several multimedia (images, audio, videos, animations ...) to present their ideas, enhancing arts and creativity in their academic.

Content Creation (CC): This component refers to both handwritten and native digital data created by students during their academic activities. Handwritten data, such as text and pictures, can be converted into digital data. The value of this data is known through integrating with third-party applications for their personalised learning plan. Content creation encompasses various forms of data, including blog posts, forum discussions, graphic assets, and text in local languages.

Artificial Intelligence (AI): A comprehensive understanding of programming and data is essential to understand the AI system. The content created by students is used to train an AI system for personalised learning. These systems will help students identify areas for improvement in their study plans.

Application-Based Learning (ABL): This component emphasises personalised learning so that students can explore their interest areas- Program, Graphics, Content Creation, or Artificial Intelligence.

Flexible Curriculum: Curriculum designers can allocate weightage to CS key components depending upon the social context. Computational thinking (CT) and Application-based learning (ABL) are the mandatory components in this framework. For example, Figure 7 presents the weightage distribution for the CS components- CT (10%), ABL (20%), Program (25%), Graphics (20%), CC (10%), and AI (15%).

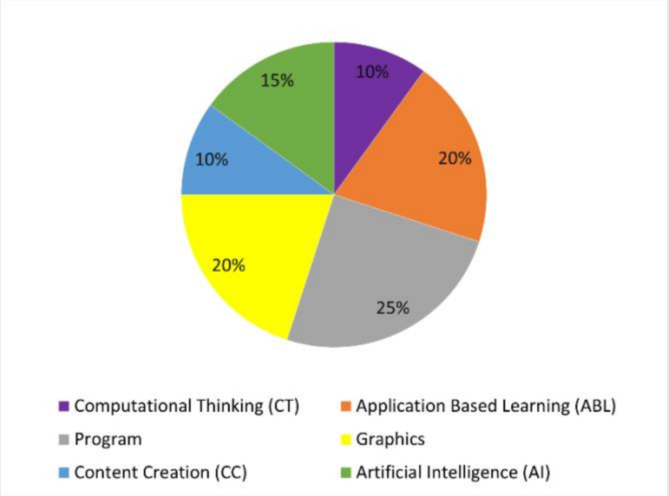

Figure 7. Flexible Curriculum

Personalised Learning: The CS curriculum development framework enhances a personalised learning experience. Students can combine the ABL component with their areas of interest. For example, if a student is interested in Graphics, then they can pursue the ABL component in Graphics. Thus, the total weightage for personalised learning becomes 40% by summing Graphics (20%) and ABL (20%) components.

7.1 CS Curriculum Development Committee and Its Roles

This study proposes a CS curriculum development committee comprising key stakeholders with their roles to develop the CS Curriculum (see Figure 8).

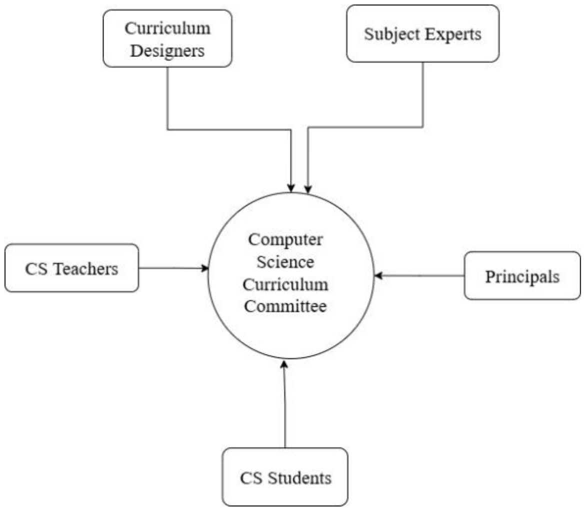

Figure 8. Structure of the CS Curriculum Development Committee

Curriculum Designer: The curriculum designer will lead the development of the CS curriculum and learning outcomes aligned with the education policies of the state, with specific attention to:

> *Readability:* The language used in the textbooks and learning resources is understandable as per the student's grade level in their social context.
>
> *Interdisciplinary*: CS concepts are presented from interdisciplinary subjects such as Math, Science, and Arts to have a comprehensive understanding of applications. The interdisciplinary knowledge is applied to CS components- Program, Graphics, AI, and ABL.
>
> *Content Organisation*: The easy/ difficult units and theory/ practical learning are balanced throughout the academic session.
>
> *Credit Weightage*: The CS curriculum is balanced with horizontal subjects in terms of credit weightage and teaching hours (theory and practical).
>
> *Print Guideline*: The whitespace characters- Space, Tab, Indent have syntax/ semantic meaning in programming and impact the program execution. There are print guidelines for CS textbooks to identify whitespace characters.

Subject Experts: Textbook authors are chosen from a committee of subject experts. They formalise the curriculum into the textbook contents from the provided topics by curriculum designers. The digital textbook is required to provide examples from the social context.

School Principals: They assess school-wise institutional resources (human and physical) required to implement the CS curriculum in their social context.

CS Teachers: They assess students' interests and ability in relation to CS components- Graphics, Program, CC, AI, ABL for personalised learning.

Students: Grades 9 and 10 students provide feedback on the textbook content’s readability and understandability, and examples relevant to their social context. Grades 11 and 12 students can provide feedback on the knowledge and skills learned from the CS subject.

8. Conclusion

Developing a CS curriculum is a complex process as it requires input from multiple stakeholders from several areas due to innovations in technologies and changes in lifestyles, society, and workforce diversity. Development of a foundational CS curriculum that delivers core concepts, knowledge, and skills through flexible and personalised learning, giving preference to the students' ability and interest within their social context, remains a key focus. The existing CS curriculum does not adequately provide an inclusive curriculum that prioritises the needs of individual students' abilities and interests. It considerably lacks the flexibility to address students' diverse interests and abilities and varying levels of school infrastructure, thereby limiting its effectiveness in providing students with adequate knowledge and skill-based education. The study is limited to the CS curricula and textbooks for grades IX and X in Nepal.

To address these shortcomings, a student-centric CS curriculum development framework is suggested, focusing on students' interests and key areas within the CS subject. The framework incorporates CS concepts from foundational concepts with an unplugged approach to a plugged approach and application-based learning as per students' interests in their domains that can be used in real-world scenarios. The framework also suggests that the curriculum development committee be composed of both curriculum development experts and students to ensure that the revised curriculum effectively addresses learning challenges and prepares students for future careers.

References

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A Revision of Bloom's Taxonomy of Educational Objectives*: *Complete Edition*. New York: Longman.

Arce-Trigatti, A., Jorgensen, S., Sanders, J. R., Kaller, H., & Arce, P. E. (2019). *The promotion of a revised TPACK Model (TSPACK): Lessons learned from the foundry inspired Steelcase active learning space project*. In Proceedings 2019 ASEE Annual Conference and Exposition.

Bonk, C. J., & Cunningham, D. J. (2012). *Searching for learner-centered, constructivist, and sociocultural components of collaborative educational learning tools*. In Electronic collaborators (pp. 25-50). Routledge.

Braun, V., & Clarke, V. (2006). *Using thematic analysis in psychology*. Qualitative Research in Psychology, 3(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Brougham, D., & Haar, J. (2018). *Smart Technology, Artificial Intelligence, Robotics, and Algorithms (STARA): Employees' perceptions of our future workplace.* Journal of Management & Organization, 24(2), 239–257. https://doi.org/10.1017/jmo.2016.55

Central Board of Secondary Education. (2023a). *Senior Secondary School Curriculum 2023-24*. CBSE.

Central Board of Secondary Education. (2023b). *Computer Applications 2023-24*. CBSE.

Comer, D. E., et al., (1989). *Computing as a discipline.* Communications of the ACM, 32(1), 9-23. https://doi.org/10.1145/63238.63239

Curriculum Development Center (CDC). (2019a). *Secondary education curriculum for elective Computer Science subjects* (Grades 9 and 10)], 137-148.

Curriculum Development Centre (CDC), (2019b). *National curriculum framework (NCF) for school education in Nepal*.

Ezeamuzie, N. O., & Leung, J. S. C. (2022). *Computational thinking through an empirical lens: A systematic review of literature.* Journal of Educational Computing Research, 60(2), 481–511. https://doi.org/10.1177/07356331211033158

Flesch, R. (1948). *A new readability yardstick*. Journal of Applied Psychology, 32(3), 221–233. https://doi.org/10.1037/h0057532

George, D., & Mallery, P. (2016). *IBM SPSS statistics 23 step by step: A simple guide and reference*. Routledge.

Gharti, S. K., Upreti, S., & Thapa, B. (2020a). *Computer Science Grade 9 (2020 AD)*. Government of Nepal, Ministry of Education, Science and Technology, Curriculum Development Center. Janak Education Materials Centre Ltd. *http://lib.moecdc.gov.np/elibrary/pages/view.php?ref=3668*

Gharti, S. K., Upreti, S., & Thapa, B. (2020b). *Computer Science Grade 10 (2020 AD)*. Government of Nepal, Ministry of Education, Science and Technology, Curriculum Development Center. Janak Education Materials Centre Ltd. https://learning.cehrd.gov.np/pluginfile.php/259/mod_resource/content/1/compscience10.pdf

Hanna, J. L. (2008). *A nonverbal language for imagining and learning: Dance education in K–12 curriculum. educational researcher*, 37(8), 491–506. https://doi.org/10.3102/0013189X08326032

Israel, M., Chung, M. Y., Wherfel, Q. M., & Shehab, S. (2020). *A descriptive analysis of academic engagement and collaboration of students with autism during elementary computer science*. Computer Science Education, 30(4), 444–468. https://doi.org/10.1080/08993408.2020.1779521

K–12 CS. (2016). *K-12 Computer Science framework*. http://www.k12cs.org

Kelly, A. V. (2004). *The curriculum: Theory and practice* (5th ed). SAGE.

Kuzu, T. S., & Durna, C. (2020). *The effect of intelligence and mind games on secondary school students' writing success.* The Turkish Online Journal of Educational Technology, 19(3).

Matthew, U. O., Kazaure, J. S., & Okafor, N. U. (2021). *Contemporary development in e-learning education, cloud computing technology & internet of things.* EAI Endorsed Transactions on Cloud Systems. https://doi.org/10.4108/eai.31-3-2021.169173

Ministry of Education, Science & Technology (MoEST). (2019). *Nepal Rastriya shiksha niti*.

Ministry of Education Bhutan (MoEB). (2021). *National School Curriculum Information & Communication Technology (ICT) Curriculum Framework Classes PP-XII.*

Ministry of Education Pakistan (MoEP). (2009). *National Curriculum for Computer Science Grades IX-X And XI-XII*.

Ministry of Law, Justice, and Parliamentary Affairs (MoLJPA). (2015). *Nepal Constitution*.

Mishra, P., & Koehler, M. J. (2006*). Technological pedagogical content knowledge: A framework for teacher knowledge.* Teachers College Record, 108(6), 1017-1054. https://doi.org/10.1111/j.1467-9620.2006.00684.x

Neupane, P. (2020). *Policy framework for education development in Nepal*. International Education Studies, 13(1), 89-97. https://doi.org/10.5539/ies.v13n1p89

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2019). *Development of computational thinking, digital competence and 21st century skills when learning programming in K-9.* Education Inquiry, 11(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844

Pinar, W. F. (2019). *What is curriculum theory?* Routledge.

Powers, K. D. (2003). *Breadth-also: A rationale and implementation*. ACM SIGCSE Bulletin, 35(1), 243-247. https://doi.org/10.1145/792548.611979

Salas-Pilco, S. Z., Xiao, K., & Oshima, J. (2022). *Artificial Intelligence and new technologies in inclusive education for minority students: A Systematic Review.* Sustainability, 14(20), 13572. https://doi.org/10.3390/su142013572

Saloviita, T. (2020). *Attitudes of teachers towards inclusive education in Finland*. Scandinavian journal of educational research, 64(2), 270-282. https://doi.org/10.1080/00313831.2018.1541819

Siemens, G. (2005). *Connectivism: A Learning Theory for the Digital Age*.

Soto, S. T. (2015*). An Analysis of curriculum development. theory and practice in language studies*, 5(6), 1129. https://doi.org/10.17507/tpls.0506.02

Stetter, M. E. (2018). *The use of technology to assist school-aged students with high incidence special needs in reading*. Education Sciences, 8(2), 61. https://doi.org/10.3390/educsci8020061

Van Laar, E., Van Deursen, A. J. A. M., Van Dijk, J. A. G. M., & De Haan, J. (2020). *Determinants of 21st-Century Skills and 21st-Century digital skills for workers: A Systematic Literature Review*. SAGE Open, 10(1). https://doi.org/10.1177/2158244019900176

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). *Computer science in K-12 school curricula of the 2lst century: Why, what and when?* Education and Information Technologies, 22(2), 445–468. https://doi.org/10.1007/s10639-016-9493-x

Weintrop, D., & Wilensky, U. (2019). *Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms*. Computers & Education, 142, 103646. https://doi.org/10.1016/j.compedu.2019.103646

William, A. (2019). *Teaching in a Digital Age—Second Edition*.

Wilson, D. D. (2011). *Applying mass customization concepts to core courses: Increasing student-centered customization and enabling cross-functional integration*. Decision Sciences Journal of Innovative Education, 9(1), 81-99. https://doi.org/10.1111/j.1540-4609.2010.00293.x

Yadav, A., Hong, H., & Stephenson, C. (2016). *Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms*. TechTrends, 60, 565-568. https://doi.org/10.1007/s11528-016-0087-7

Zhansulu, S., Batima, T., Temirov, K., Shalgynbayeva, K., Shamurat, O., & Mukhamedzhanov, B. (2022). *Supporting children with special educational needs as a socio-pedagogical problem.* Cypriot Journal of Educational Sciences, 17(7), 2262-2273. https://doi.org/10.18844/cjes.v17i7.7692

Zhou, S., Jeong, H., & Green, P. A. (2017). *How consistent are the best-known readability equations in estimating the readability of design standards*? IEEE Transactions on Professional Communication, 60(1), 97–111. https://doi.org/10.1109/TPC.2016.2635720

International Journal of Computer Science Education in Schools, September 2025, Vol. 7, No. 2
1ISSN 2513-8359

Learning to Compute:

An Overview of Computing Education in England from 1970 to 2014

Dr Bea Wohl

b.wohl@arts.ac.uk

[1]University of the Arts London, Creative Computing Institute

DOI: 10.21585/ijcses.v7i2.231

Abstract:

This paper will present a brief history of computing education in England from 1970 to 2014. It sets out to provide the context which shaped the 2014 computing curriculum. After this curriculum had been in place for almost a decade, the paper provides an opportunity to see how computing skills, including information communications technology (ICT), have been taught in England and how that shifted towards a more computer-science-focused curriculum in 2014. This paper gives a brief overview of how the formal subjects of information technology (IT), ICT, and computing became a strategic part of UK compulsory education. It examines the introduction of computers into schools in the 1980s, what computing in schools looked like, and how different forms of computing were variously integrated into the national curriculum for England and Wales.

Keywords: computing curriculum, ICT, Digital literacy, Digital Economy, Computing education

1. Introducing the History, Influence and Themes of the Computing Curriculum

This paper sets out to provide a brief introduction to the forces that shaped the 2014 computing curriculum and to present a brief history of what computing education in England looked like prior to that point. The paper is a narrative review of computing education literature, compiled as part of the author's doctoral studies, which builds on this and other literature to conduct qualitative research with young people about the impacts of the 2014 computing curriculum (Wohl, 2020). The purpose of this review of literature is to provide a historical accounting of the development of Computing in primary and Secondary education, providing a foundational context for past, present and future research (Baumeister & Leary, 1997). While many aspects of this history have been compiled in other papers and publications, previously, no other paper has brought together in a single narrative this development of teaching computing in the UK from its earliest beginning to when Computer Science was codified into the English national curriculum in 2014. Many of the authors cited throughout have provided key aspects of this exposition (Young, 1991; Woollard, 2018; Savage & Csizmadia; 2018; Passey, 2014, 2015, 2016; Anderson,

& Levene, 2012), whereas other literature provided context, background, and a theoretical lens through which to understand these changes to how this key topic was delivered.

2. A Brief History of Computing and IT in English Schools

Formally, the teaching of computing in schools started in the 1970s with the addition of "computer studies to the GCE (A and O level) examinations for 14–18-year-olds" (Woollard, 2018, p. 14). This addition enabled pupils to approach computing as a potential career and a subject for study in higher education. The UK government began to see competing in the information economy as essential for economic success, and a skilled workforce as central to this aim. Mathematics teachers started to integrate computing technology with logical problem-solving, and teachers across the curriculum recognised the potential of computers for their subjects as well (Passey, 2014). The potential future impact of computers was further highlighted in 1978 with the broadcast of the Horizon television programme, Now the Chips Are Down, which made dramatic predictions about how the transformative technology of computers and microchips would replace human jobs (Anderson & Levene, 2012). While too dire, these predictions highlighted the possibility that computers would powerfully drive the future economy and that individuals would need to learn the associated skill sets to remain relevant.

By the late 1970s, research machines (RM) had developed microcomputers for schools, with the Apple II and the Commodore PET released around the same time for the education market (Passey, 2014). Due to the high cost of machines such as the RM 380Z, however, most schools could not invest in the new technology. The cost of computers fell in the late seventies and early eighties once UK companies began mass-producing them, such as Sinclair's personal computer with a QWERTY keyboard for less than £100 (Anderson & Levene, 2012; Haddon, 1991). Multiple government programs, such as the Department of Education and School (DES) and the Department of Trade and Industry (DTI), launched initiatives to fund computers and computing in schools at this time (Passey, 2014).

By 1986, education was seen as the dominant factor in ensuring Britain's success in the worldwide information society (Lyon, 1991). Microelectronics and computing became essential to the UK's economic success, and technology became a necessary part of both teaching and learning (Linn, 1991). Still, a great deal of mixed messaging circulated about the 'purpose' of this new subject. While some lessons focused on problem-solving and computer programming, others focused on the use of specific applications. For example, the provision and inclusion of computing in primary schools was ad-hoc on a county basis, hampered by a lack of hardware, a lack of training and mostly a lack of adequate software to meet educational needs (Jackson et al., 1986). Jackson et al. (1986) also highlight a concern that including computers in this classroom would have adverse social effects on pupils. By the introduction of the UK's first national curriculum in 1989, the focus in computing had shifted to general computer usage (Woollard, 2018). The 1989 national curriculum came in Margaret Thatcher's third term in government, from a place of deep concern with the standard of education in the UK, a desire for a higher standard across the board (Whetton, 2008). While the new curriculum did not specifically include computing, it encouraged schools and teachers to implement information technology across the curriculum. Not until the 1998 curriculum reform would every pupil learn IT (Passey, 2014).

3. The ‘Golden Age’: Computing in Schools in the 1980s

While computing as a school subject got off to a slow start, the 1980s saw an explosion of interest and funding. Exemplary of the hype and promise early in the decade, the BBC released the BBC Micro accompanied by a series of television broadcasts called the Computer Literacy Project5. The idea was that children who knew how to code would have a significant advantage over those who did not, as reinforced by news stories about ‘whiz kids’ making a living from computer programming (Anderson & Levene, 2012). The BBC was not alone in investing resources into children’s coding; other television programs broadcast in the early eighties also brought the idea of computers and programming into British living rooms (Anderson & Levene, 2012; Webster et al., 1991).

In the United Kingdom, on a national level, 1982 saw the launch of the conservative government’s multi-million-pound campaign, ‘IT82’, to raise awareness of the value and power of information technology. This campaign focused on the potential of technology to transform society while encouraging the public, especially young people, to learn how to use it (Webster et al., 1991). By the mid-1980s, schools began to demand national funding to invest in equipment, technology, and training. The DTI provided significant school funding that prioritised hardware manufactured in the UK, such as the RML 380z, BBC Acorn, and the Sinclair Spectrum (Linn, 1991; Passey, 2014). By the end of the decade, DTI funding had procured computers, disk drives, mice, monitors, Teletext adaptors, telephone lines, and Turtle robots. By the early nineties, most schools had computer networks that linked them with other schools (Passey, 2014). While this was true in secondary schools, provision in primary schools was still patchy and dependent on teacher attitudes and training when it came to actual use (Hermans et al., 2008).

While the DTI was funding hardware, the DES and the Department of Employment were funding the Microelectronics in Education Programme and the Technical and Vocational Education Initiative, respectively, to train teachers and prepare young people for a future society where computers were commonplace (Linn, 1991; Passey, 2014). As a result of additional resources such as educational videos and teaching materials produced by the BBC as part of ‘Micros in Schools’, the BBC Micro became the computer of choice for many schools (Anderson & Levene, 2012). But computers were available mainly in computer labs, and non-IT subject teachers found they had limited access to technology-based resources in the classroom such that, ‘from 1989, the idea of IT “across the curriculum” was felt […] to be worthwhile, but practically difficult to achieve’ (Passey, 2014).
With the approaching publication of the national curriculum in 1989, the emphasis on IT shifted to the enrichment of the entire curriculum with topic-based resources and software (Passey, 2014). The arrival of the Microsoft Windows operating system made multimedia tools accessible for schools and young people, producing an uptick in the authoring of multimedia technologies as a core part of IT (Woollard, 2018). At the launch of the national curriculum, the pressure to use computing and IT across subjects became explicit with the stated aim that every pupil uses IT to enhance learning in every subject (Passey, 2014) and specifications for what this should look like (Woollard, 2018). A large amount of money and excitement rapidly increased the opportunities for pupils in the 1980s to learn about and with computers. Much of the early focus in the 1980s was on learning computer-specific skills such as logic and programming. By the end of the decade, ‘IT82’ was all but forgotten, and the focus shifted to using computers rather than learning how they worked or how to program them (Passey, 2016).

[5] The archive of this material can still be found at https://clp.bbcrewind.co.uk/ (accessed 15/4/2025)

4. English National Curricula in 1989, 1998, and 2004

1989 was a watershed moment in which the first national curriculum significantly changed the education system across the country, specifying 'subject content […] for all sectors of compulsory education [with a statutory] learner entitlement curriculum' for the first time (Passey, 2014). To a greater or lesser extent, technology use was integrated into every subject area at every age level, although the primary focus remained 'design technology' (Passey, 2014). Despite not having its own curricular area, IT received five strands of progression called 'IT capability to be used for assessment': developing ideas and communicating information, handling information, modelling, measurement and control, application and effects (Barnes & Kennewell, 2018; Passey, 2014). This version of the national curriculum lasted for a decade, seeming to stabilise, formalise, and institutionalise how IT was used in schools to the point of stagnation.

With the change of government in 1997, ministers began to discuss the use of IT for broader concerns about learning, attainment, and effectiveness to raise standards across all subject areas (Passey, 2014). The 1997 Stevenson report raised concerns regarding basic IT confidence and competence among both teachers and pupils, as well as the need for a stable policy on computing learning (Woollard, 2018). This report came alongside a growing debate regarding New Labour's approach to education more generally (Whetton, 2008). Stevenson's concerns laid the groundwork for the 1999 curriculum reform, which revisited IT to create the discrete subject of information communications technology (ICT). This new subject had its own five components that defined 'ICT capability':

- Routines such as using a mouse or double-clicking on an application.
- Techniques such as adjusting margins to make text fit a page.
- Key concepts such as menu, file, database, spreadsheet, website, and hypertext link.
- Processes such as developing a presentation, seeking information, and organising, analysing and presenting the results of a survey.
- Higher-order skills and knowledge, such as recognising when the use of ICT might be appropriate, planning how to approach a problem, making and testing hypotheses, monitoring progress in a task, evaluating the result, and reflecting on the effect of using ICT in a particular situation. (Barnes & Kennewell, 2018)

These five capabilities moved the curriculum away from programming toward applying ICT to specific, often work-related tasks. Computers have moved from being 'new technology' for exploration and investigation to essential tools for any future career a child might imagine. While many concepts associated with computer science do appear among the five components, they bunch into the final capability of choosing the right tool for the right task rather than making one's own tools (Woollard, 2018).

By creating a discrete subject area for ICT, the 1999 curriculum made time for the teaching and learning of ICT but also took emphasis away from its use across the curriculum. By 2002, the focus once again came to computer suites in primary schools and specialist teaching in secondary schools (Woollard, 2018). At this time, the greatest challenge to computing across the curriculum was teachers' reluctance to use emerging technology to the point that IT skills were only taught and utilised in IT lessons. By 2004, the secondary school strategy brought about the 're-emergence of the ICT across the curriculum' (Woollard, 2018). While there was relative stability in IT provisions after the initial national curriculum from 1989 to 1997, it raised questions on computers in schools, such as what should be taught (programming, IT, ICT); why should it be taught (for industry, for work, for any other reason); and should it be discrete and or cross-curricular?

5. Where Did the 2014 Computing Curriculum Originate?

When examining the context for the 2014 computing curriculum, it's worth reflecting on Young's observation from 1991 that discussions of technology in education are polarised between specialists with knowledge of electronics and computing and 'the rest who have little concrete knowledge of either (or any other) technologies' (Young, 1991). Throughout the process of curriculum development, this tension was apparent. A range of experts advised on the changes with often contradictory recommendations that were then implemented by nonexperts (Williamson, 2015). There was at least consensus that the subject of computer science should be (re)introduced into the compulsory curriculum in the 2014 version (Passey, 2016). Computer science was a relatively new addition to post-compulsory education, with the first computer science departments arising in universities in the early 1960s and the first doctor of computer science awarded in 1965 (Passey, 2016). This is worth noting for two reasons: first, the knowledge that constitutes computer science is not fixed nor clear to the general population the way a subject such as mathematics might be and second, university student applications for computer science courses had begun to decline prior to 2014 (Peyton-Jones, 2009).

By this point, the pre-sixth-form compulsory curriculum in England emphasised ICT literacy for a small number of computer applications (Peyton-Jones, 2009). The more universal the computer became, the more 'the attractiveness of learning ICT had seemed to decrease' (Peyton-Jones, 2009). Much of the intended revisions to the computing curriculum reinvented the topic so it could appeal to a broader range of students; computing could apply to every aspect of life and almost any school subject (Peyton-Jones, 2009). To raise awareness for the new computing curriculum, the UK government established the 'Year of Code' campaign in 2014, linking the ability to write computer programs to potential economic success (Williamson, 2015), an effort that paralleled the aforementioned IT82 initiative.

6. CAS, Restart, and NextGen: Voices Shouting for Change

Several factors influenced the 2014 computing curriculum, including grassroots efforts and two independent reports, with the final catalyst for change being a speech by a global industrialist. In 2008, a group of academics, industrialists and teachers came together to form Computing Next Generation (CNG), an organisation that later evolved into the group Computing at School (CAS) (Woollard, 2018). CAS started as a newsletter and, over time, brought together a range of individuals frustrated by the lack of computer science in the national curriculum

(Davies, 2017). CAS membership comprised: enthusiastic teachers with the freedom to include some aspects of computer science in their lessons; academics concerned with the drop in applications to study computer science programs at UK universities; and industry representatives from large technology companies concerned by the lack of higher-level computer science knowledge across the population (BCS, 2010; Peyton-Jones, 2009; Sentence & Csizmadia, 2015; Williamson, 2015; Woollard, 2018). CAS received funding from the British Computing Society (BCS), also known as the Charter Institute for IT, and by 2010, started to publish white papers (BCS, 2010; CAS, 2010; CAS Working Group, 2012; Peyton-Jones, 2009), including a proposed rethink of how to teach computing in UK schools (Williamson, 2015).

In early white papers, CAS distinguished ICT as a set of skills from computer science as a discipline. CAS was critical of the focus on ICT, making the case that while ICT was important, it was equally important that every child be introduced to computer science, even if not every child would make use of the concepts which comprise it (Peyton-Jones, 2009). CAS's official position was that pupils were no longer learning enough computing because the 'exclusive emphasis on ICT means that today's school pupils have fewer opportunities to learn Computing than they did 20 years ago' (Peyton-Jones, 2009). CAS drew a connection between decreasing computing enrolments and problems with recruitment for computing careers. They blamed this change on the key stage 3 curriculum, which was associated with learning how to use 'office-type' software but lacked opportunities for creativity (for example, through programming) and learning how computers work (Woollard, 2018). This could also be connected with ICT becoming a discrete subject, removing the opportunity for creativity and integration across the curriculum. Another challenge may have been the lack of ICT/computer specialists teaching in secondary schools.

While CAS was advocating for a more comprehensive approach to computer science in schools, two public reports reviewed the state of ICT in schools. The 'Next Gen' report, published by Nesta and written by Livingston and Hope(2011), looked at the connection between the computing skills young people learned in school and the digital and creative economy more broadly. Specifically, Livingston and Hope(2011) examined whether young people were learning the skills and concepts they needed to enter the UK video game and visual effects industries. This report advocated for young people to gain a rigorous knowledge of computers and code and advanced computer science as central to the national curriculum (Livingston & Hope, 2011; Williamson, 2015). The authors also argued the importance of linking computing with the arts in UK schools (Livingston & Hope, 2011).

In 2012, the Royal Society, the UK's independent scientific academy, released a report calling for the reintroduction of computer science to UK schools, harkening back to when the BBC Micro introduced a generation to programming (Royal Society, 2012; Williamson, 2015; Woollard, 2018). Entitled 'Shut Down or Restart? The Way Forward for Computing in UK Schools,' the report was written by Steve Furber, a member of the team which initially developed the BBC Micro (Anderson & Levene, 2012; Royal Society, 2012) and commissioned by Microsoft, Google, and many university computer science departments (Williamson, 2015). 'Shut Down or Restart' criticised the state of ICT in UK schools, stating, 'current delivery of Computing in UK schools is highly unsatisfactory' primarily because:

1. The current national curriculum in ICT may be broadly interpreted and reduced to the lowest level when non-specialist teachers deliver it.
2. There is a shortage of teachers who can teach beyond basic digital literacy.
3. There is a lack of continuing professional development for teachers of computing and
4. Features of school infrastructure inhibit effective teaching of computing (Royal Society, 2012)

While the 'Next Gen' and the 'Shut Down or Restart?" reports as well as CAS's ongoing work were instrumental to the revision of the computing curriculum, the catalyst is generally considered to be Eric Schmidt, then head of Google, in his 2011 MacTaggart lecture (Woollard, 2018). The speech mobilised political support for computing reform, highlighting some of the same conclusions as the two public reports (Williamson, 2015). Schmidt (2011) used this platform to express dismay at the state of computer science in UK schools, stating it was hard for Google to find the talent they needed in the country (Cave & Rowell, 2014; Williamson, 2015). Schmidt (2011) metaphorically threw down a gauntlet for the UK government by stating, 'Your IT curriculum focuses on teaching how to use software but gives no insight into how it's made'. Just as the Next Gen report had called for integrating computer science with the arts (Livingston & Hope, 2011), Schmidt (2011) challenged the UK to 'bring art and science back together'.

These comments prompted politicians to rebuke the teaching of ICT and call for more computer science (Williamson, 2015). Michael Gove (2012), then minister for education, gave a speech at the BETT exhibition in 2012 in which he called ICT "harmful, boring and/or irrelevant" and promised to reform the teaching of computing in UK schools with a rigorous approach to computer science in the new national curriculum. With input from Nesta, BCS, Google, Microsoft, CAS, and Raspberry Pi (Williamson, 2015), Gove (2012) made it clear that the goal of the new computing curriculum was to move away from ICT and towards computer science.
While various forces made computer science more appealing than ICT to schools and young people, most (except Michael Gove's) called not for the replacement of skills taught in ICT but rather a supplementation with computer science. Akin to Young's (1991) observation on specialists and non-specialists influencing the use of technology in schools, specialists in this case called for a curriculum that included computer science but also drew on digital literacy, ICT skills, and the arts. More importantly, the "Shut Down or Restart?" report stated, "Computer Science is sufficiently […] foundational that it should be recognised as a high-status subject in schools, like mathematics, physics or history" (Royal Society, 2012). Taking this statement seriously would have meant giving the subject of computing a much larger portion of a child's school day. Instead, while the content changed, the amount of time devoted to computing at most secondary schools remained constant: approximately one hour per week (Kemp & Berry, 2019).

7. Why So Much Computer Science?

A range of arguments have been made for why the ICT curriculum is no longer fit for purpose. On the one hand, CAS had argued that ICT provisions, while serving some students well, hampered more able students (Peyton-Jones, 2009). On the other hand, a move towards computer science moves from a skill to a discipline which prepares "young people for jobs that don't yet exist, requiring technologies that have not yet been invented, to solve problems of which we are not yet aware" (Peyton-Jones, 2009). However, both arguments pertain to the most academically able students. Both fail to ask: What about students who need to do jobs that already exist? Or use technology that has already been invented? Or solve problems we as a society are already all too aware of? University lecturers often report that students find computer science a hard subject (Passey, 2015; Passey, 2016). Considering that computing, the 'harder' subject, was required to fit within the same timetable as ICT, it seems self-evident that many pupils would find it more challenging to make progress. Moreover, many ICT skills remain important for pupils' futures, and a complete shift from ICT to computer science might mean ICT competencies would be lost. Rather, some sort of balance was necessary instead of a wholesale pivot (Passey, 2015). But the question remained whether schools had the facilities to grant teachers access to technologies supporting both ICT and computer science (Passey, 2015).

The argument of "not one nor the other but both" closely mirrors the conclusions of the "Shut Down or Restart?" report from the Royal Society (2012). The wording of these findings is key: computer science is as important as any other science; every child should be 'digitally literate', and digital literacy is as important as reading and writing; children should have the opportunity to learn the concept and principle of computing; and there should be opportunities to take these topics forward in a range of ways. From the phrasing alone, it seemed the two most important aspects would be to devote more time to computing lessons and to ensure every child was digitally literate by providing ample opportunities in which pupils could choose to engage or not.

8. The Case for a New Curriculum

A CAS white paper from 2009 made the case that, in fact, much could have been achieved with the 'admirably non-prescriptive' KS3 curriculum in terms of computing, such as 'planning, testing and modifying a sequence of instructions, recognising where a group of instructions needs repeating and automating frequently used processes by constructing efficient procedures that are fit for purpose' (Peyton-Jones, 2009). If so, then the circumstance for change was not a new curriculum but rather a new approach to the existing curriculum. In contrast, Williamson(2015) contends the new curriculum meant to embed a new way of thinking about the world; in fact, computational thinking and thinking through code would change how young people thought about the world and the phenomena they encountered that might be described in code and understood as computable. Williamson(2015) believed it would benefit large technology companies and government systems if pupils understood the world through computers. An overarching question, then, was whether a new curriculum would solve the problems seen as inherent to the ICT curriculum that were depressing the numbers of computer science majors and professionals (Woollard, 2018).

9. 'Computing': 'I Don't Think That Word Means What You Think It Means'

The shift from ICT to computing moved away from skills and towards disciplinary principles and ideas, ways of understanding problems and describing the world (Peyton-Jones, 2009). Disciplinary language allowed CAS to argue that computing takes its rightful place alongside mathematics and the other sciences (Peyton-Jones, 2016). No longer would students learn only how to use office-type software; they would also learn a discipline which would prepare them to understand the world they lived in better. The move towards computing reflected an international change where several countries introduced elements of computing into schools as either formal or informal learning opportunities (Savage & Csizmadia, 2015).

The idea behind the focus on computing was that it would create a more ‘exciting’ subject than ICT. Seeding language for Michael Gove’s speech two years later, CAS went so far as to state: ‘Computing is one of the most exciting subjects on earth. Yet the current arrangements for teaching computing concepts at school leave many of our students feeling that it is irrelevant and dull’ (CAS, 2010). Given this mixed reception, what exactly made up ‘computing’? It focused on theory rather than practice—concepts rather than artefacts (CAS, 2010)—based on the following four points:

- The study of algorithms and data structures: efficient and ingenious ways to solve computational problems, together with a rich underlying theory of the ‘complexity’ of such algorithms.
- An understanding of computer systems and networks: for example, how the internet works, and the protocols that keep data flowing smoothly despite all the control being decentralised.
- An appreciation of the challenges of human-computer interaction, which focuses on the challenge of making computers accessible to people.
- How computers work. Traditionally, this means gates, binary arithmetic, and digital hardware. More broadly, however, biologically inspired computation paradigms are in rapid development. (CAS, 2010; Peyton-Jones, 2009)

Interestingly, these points do not include using specific software or learning programming or coding. What can be concluded is that the discipline of computing focused on a theoretical understanding of the digital world and was expected to be more exciting than learning how to use software. Ultimately, the purpose of computing as opposed to computer science is to provide students with skills and knowledge that can be applied to solutions across STEM subjects (BCS, 2010). Therefore, computing is cross-curricular by nature, but it is also theoretically specific and applicable to any number of other disciplines.

10. Are They Ready for the… Digital Economy?

Even in earlier debates about IT/ICT and computing in schools, the digital economy played a central role. In the 1980s, much of the argument for increasing the role of IT in schools was tied to the rise in technical vocations and predictions that microelectronics technology would transform almost every industry. It seemed essential that pupils became computer literate in school (Mackay, 1991; Webster & Robins, 1991). The IT82 campaign claimed IT would transform every sector of the economy, from factory work to public houses, management to healthcare (Webster & Robins, 1991). In hindsight, these predictions are surprisingly prescient: almost every sector has been transformed by some form of technology, an excellent economic argument for teaching relevant computing education in schools. However, relevance does not answer the question of what should be taught.

The Royal Society (2012) report, while making the case for digital literacy, underscored the danger that the lack of enthusiasm associated with the ICT curriculum could have detrimental effects on UK's education system and the ability of the UK to compete on the international stage. The report makes a link between learning to code and entrepreneurial success, highlighting that "the two most successful start-ups in Computing and [the] business world—Facebook and Google—were led by people who had been writing software at university" (Royal Society, 2012). The growth in IT industries of roles in software engineering prompted commentators to glamorise the discipline of computing to recruit skilled workers for the digital sector (Williamson, 2015).

Livingston and Hope (2011) conjectured that the digital economy would not be based on a small set of technical skills but on multidisciplinary skills such as teamwork, communication, and artistic ability. They found that pupils had little sense of what it would be like to work in the digital economy, even in a relatively appealing sector such as the video game industry (Livingston & Hope, 2011). So, while the computing curriculum needed to give young people the skills to participate in this sector, they also needed to gain a sense of what careers in this sector might entail. Even with curriculum reform underway, there was still concern many young people were missing out on the opportunities available to them in the ever-expanding digital work. The Labour party commissioned a report, 'Digital Skills for Tomorrow's World', to determine what skills would be needed for the new economy and discovered all sectors needed IT abilities (Philbin, 2014).

One goal of including computing in the national curriculum was to equip every child with the competencies they would need as adults to make 'intelligent and informed choices about the digital technology that underpins their world and [and to make] valuable contributions to our digital society and economy'; this way, they could take their 'proper place in a digitally enabled, knowledge-based society and economy" (BCS, 2010). The case for the digital economy was clear, although the hype may have underplayed the downsides to technological careers where jobs often come with a degree of fragility, complexity, and mundanity (Williamson, 2015).

11. Looking Back at Computing Education and Thinking Forward

As in previous decades, delivering new technology to schools today is an investment. In the 1980s, multiple government agencies funded training and equipment. By 2014, the UK government was providing only £3.5 million, the equivalent of £175 per school, to deliver this new 2014 curriculum. Teachers needed access to continuing professional development to acquire subject knowledge and adapt their teaching to the new computing curriculum; however, these accommodations were slow to materialise (Philbin, 2014). The computing curriculum made the UK a world leader as the first country in the G20 to recognise the importance of teaching young people computing. This eminence has had a lasting legacy for education, the country, and the UK economy (Woollard, 2018). The lingering question is not whether this legacy is a success or failure, but rather, to what extent did and does the UK's computing curriculum meet the ongoing needs identified prior to its implementation?

12. Postscript: Moving forward from here

Since 2014, there has not been a new English curriculum, but the teaching of computing has continued. The establishment of the National Centre for Computing Education, funded by the Department of Education has started to fill the gap for training teachers (Teach Computing 2019). However, a 2020 survey of 100 Key Stage 3 and GCSE computing teachers found that teachers still found the curriculum dominated by teaching programming. While they value programming, they also feel it takes too much time (Mee, 2020). Many teachers find that with so much focus on programming, other aspects of ICT are being pushed out. In the primary school context, teachers find it challenging to implement key computing concepts when they are already trying to teach basic Maths and English in preparation for key exam points. There continues to be a lack of breadth and balance in computing (Mee, 2020), resulting in a subject that remains "inaccessible to most" (Mee, 2020). In terms of teaching programming, a key development in the research is Kallia and Sentance's (2017) investigation into "threshold concepts" in computing or concepts that are key to enabling learners' understanding of a subject or discipline (Kallia & Sentance, 2017). Understanding the threshold concepts can be essential to ensuring that teaching is effective, particularly as programming has come to dominate the teaching of computing. Working with teachers, they found that: "Arguments, calling a function, Control Flow, Parameters, Parameter passing, Procedural Decomposition and Design, Recursion, Return Values, Variable, Variable Scope, and Abstraction" are all potential threshold concepts (Kallia & Sentance, 2017). Others have recommended the integration of computational thinking into classroom activities in earlier years, deploying games, robots, and play to ensure a strong conceptual foundation (Lee et al., 2022).

There continues to be a need to understand computing education better as it is delivered. Falkner et al. (2019) have developed a tool for examining the differences between the prescribed curricula and how they are enacted. This tool is of particular use (as it develops) to compare the differences between different countries' enactment of computing curricula (Falkner et al., 2019). This tool highlights how teachers remain the gatekeepers of computing, and their interpretation of the relevant curriculum is paramount (Falkner et al., 2019). While an understanding and use of computers will remain relevant, how this is tackled in education is worth keeping track of, as the intersection informs it of public policy and technological development.

References

Anderson, M., & Levene, R. (2012). *Grand thieves et tomb raiders*. London: Aurum.

Barnes, J., & Kennewell, S. (2018). The impact of teachers' perspectives on the development of computing as a subject. In S. Younie & P. Bradshaw (Eds.), *Debates in computing and ICT* (1st ed., pp. 43-52). Oxon: Routledge.

Baumeister, R.F., & Leary, M.R. (1997). Writing narrative literature reviews. *Review of General Psychology*, 1(3), 311–320. https://doi.org/10.1037//1089-2680.1.3.311

BCS. (2010, November). *BCS, The Chartered Institute for IT in association with the Computing at School group Consultation Response to: The Royal Society's Call for Evidence - Computing in Schools* https://www.bcs.org/upload/pdf/royalsociety-response-nov10.pdf [Accessed 28 May 2019].

Bers, M., Flannery, L., Kazakof, E., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, 72, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Computing at School (2010). *Computing at School: a national working group.* http://www.computingatschool.org.uk/ [Accessed 28 May 2019].

Computing at School Working Group (2012). *Computer Science: A curriculum for school.* http://www.computingatschool.org.uk [Accessed 4 Feb. 2016].

Davies, R. (2017, January). Interview with Roger Davies re: history of Computing at School.

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole., E., Liebe, C., Maiorana, F., Mcgill, M. M., & Quille, K., (2019). An International Study Piloting the MEasuring TeacheR Enacted Computing Curriculum (METRECC) Instrument. *Proceedings Of The Working Group Reports On Innovation And Technology in Computer Science Education,*. 111-142. https://doi.org/10.1145/3344429.3372505 [Accessed 18 February 2021].

Haddon, Leslie., (1991). The cultural production and consumption of IT. *Understanding Technology In Education* (1st ed.), (pp. 1-13). London: The Falmer Press.

Hermans, R., Tondeur, J., Van Braak, J., & Valcke, M. (2008). The impact of primary school teachers' educational beliefs on the classroom use of computers. *Computers & Education*, 51(4), 1499-1509.

Jackson, A., Fletcher, B. and Messer, D.J. (1986). A survey of microcomputer use and provision in primary schools, *Journal of Computer Assisted Learning*, 2(1), 45–55. https://doi.org/10.1111/j.1365-2729.1986.tb00063.x.

Kallia, M., & Sentance, S. (2017). Computing Teachers' Perspectives on Threshold Concepts. *Proceedings Of The 12Th Workshop On Primary And Secondary Computing Education*. Nijmegen, Netherlands. https://doi.org/10.1145/3137065.3137085 [Accessed 18 February 2021].

Kemp, P. & Berry, M. (2019). T*he Roehampton annual computing education report: Pre-release snapshot from 2018*. London: University of Roehampton.

Linn, P. (1991). Microcomputers in education: Dead and living labour. In M. Hughie, M. Young, & J. Beynon (Eds.), *Understanding Technology In Education* (1st ed., pp. 1-13). The Falmer Press.

Lee, J., Joswick, C., & Pole, K. (2022). Classroom Play and Activities to Support Computational Thinking Development in Early Childhood. *Early Childhood Education Journal*. 51, 457-468(2023). https://doi.org/10.1007/s10643-022-01319-0

Livingstone, I. and Hope, A. (2011). *Next Gen.* [online] London: Nesta. https://www.nesta.org.uk/report/next-gen/ [Accessed 28 May 2019].

Lyon, D. (1991). The information society: Ideology or utopia. In H. Mackay, M. Young, and J. Beynon (Eds.), *Understanding Technology in Education* (1st ed., pp.93-108). London: The Falmer Press.

Mackay, H. (1991). Technology as an educational issue: Social and Political perspectives. In H. Mackay, M. Young, and J. Beynon (Eds.), *Understanding Technology in Education* (1st ed., pp.1-13). London: The Falmer Press.

Mee, A. (2020). *Computing in the school curriculum: a survey of 100 teachers*. https://doi.org/10.13140/RG.2.2.19883.59689 [Accessed 18 February 2021].

Passey, D. (2014). Early uses of computers in schools in the United Kingdom: Shaping factors and influencing directions In *Reflections on the history of computers in education: Early use of computers and teaching about computing in schools* (pp. 131-149). Berlin, Heidelberg: Springer Berlin Heidelberg.

Passey, D. (2016). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies*, 22(2), 421-443.

Passey, D. (2015). Computer science (CS) or information and communication technologies (ICT): the curriculum needs both. In *IFIP TC3 Working Conference A New Culture of Learning: Computing and next Generations*. 26-44.

Peyton Jones, S. (2009). *Computing at School: The state of the nation.* https://www.computingatschool.org.uk/data/uploads/CAS_UKCRC_report.pdf [Accessed 28 May 2019].

Philbin, M. (2014). *Digital skills for tomorrow's world.* Digital Skills Task Force. https://policy.bcs.org/sites/policy.bcs.org/files/Interim%20report.pdf [Accessed 28 May 2019].

Savage, M. & Csizmadia, A. (2018). Computational thinking and creativity in the secondary curriculum. In: S. Younie and P. Bradshaw (Eds.*), Debates in Computing and ICT Education* (1st ed., pp. 43-52). Oxon: Routledge.

Schmidt, E. (2011). *Eric Schmidt's MacTaggart Lecture in full*. [online] https://www.campaignlive.co.uk/article/edinburgh-tv-festival-eric-schmidts-mactaggart-lecture-full/1087474?src_site=mediaweek [Accessed 26 May 2019].

Selby, C. (2015). Relationships computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *WiPSCE '15 Proceedings of the workshop in primary and secondary computing education* (pp. 80-87). http://dx.doi.org/10.1145/2818314.2818315

Sentence, S. & Csizmadia, A. (2015). Teachers' perspectives on successful strategies for teaching Computing in school. In *IFIP TC3 Working Conference 2015 :A New Culture of Learning: Computing and next Generations.* 243-252. ISBN 9786099576008

Teach computing. (2019). https://teachcomputing.org/

The Royal Society (2012). *Shut down or restart? The way forward for computing in the UK schools.* [online]. https://royalsociety.org/topics-policy/projects/computing-in-schools/report/ [Accessed 28 May 2019].

Webster, F., & Robins, K. (1991). The selling of new technology. In M. Hughie, M. Young, & J. Beynon (Eds.), *Understanding Technology In Education* (1st ed., pp. 66-92). The Falmer Press.

Williamson, B. (2015). Political computational thinking: Policy networks, digital governance, and 'learning to code'. *Critical Policy Studies*, 10(1), 39-58.

Woollard, J. (2018). Computer studies, information technology, ICT, and now computing. In: S. Younie and P. Bradshaw, (Eds.), *Debates in Computing and ICT Education*, (1st ed., pp. 43 – 52).

Young, M. (1991). Technology as an educational issue: Why it is so difficult and why it is so important. In M. Hughie, M. Young, & J. Beynon (Eds.), *Understanding Technology In Education* (1st ed., pp. 234-244). London: The Falmer Press.

www.ijcses.org