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

[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.

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 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.

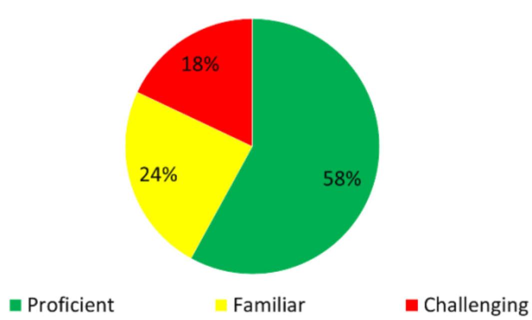

Figure 1. CS Students' Confidence in Programming Concepts

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.

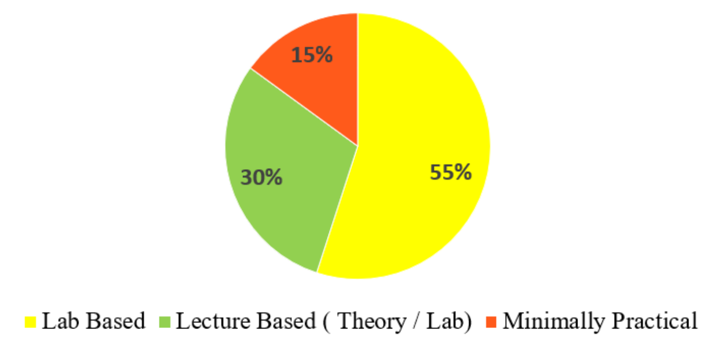

Figure 2. Teaching-Learning Pedagogy Used in CS Classes

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).

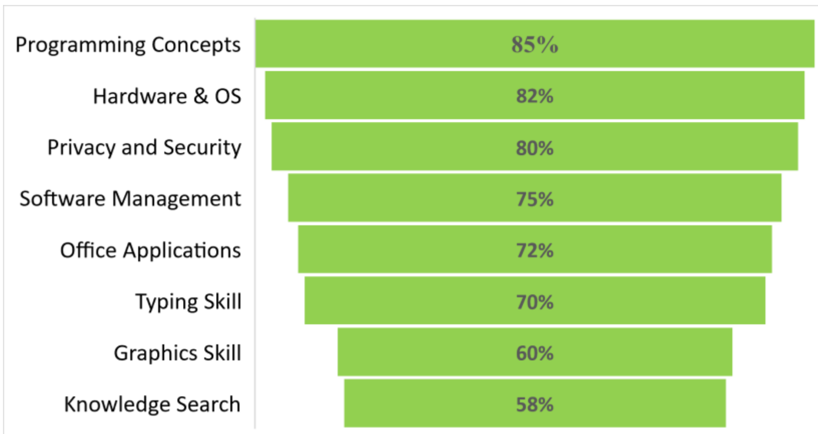

Figure 3. Expected Skills of CS Grade 10 Students

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).

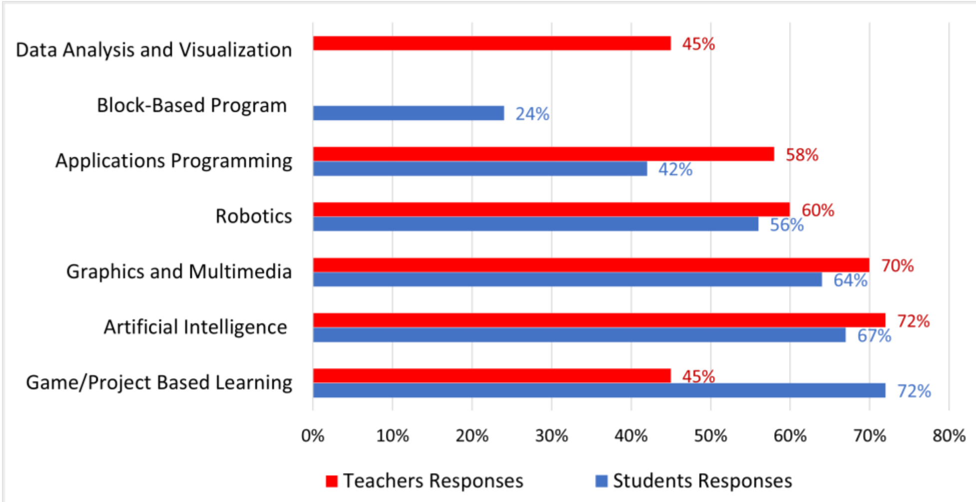

Figure 4. Students' and Teachers' Views on the Revised CS Curriculum

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).

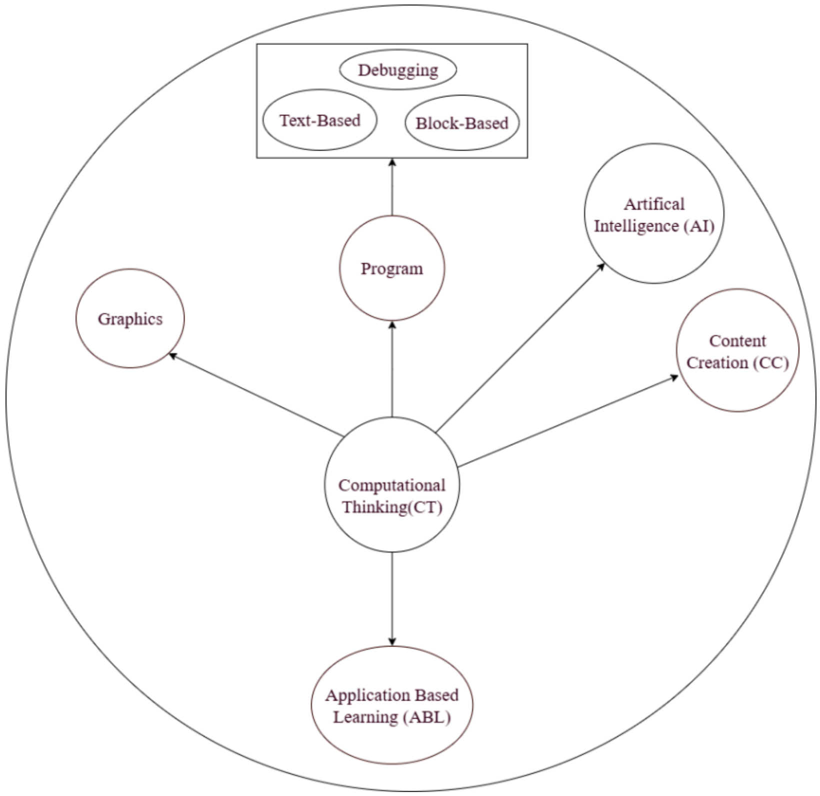

Figure 6. Student-Centric CS Curriculum Development Framework

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%).

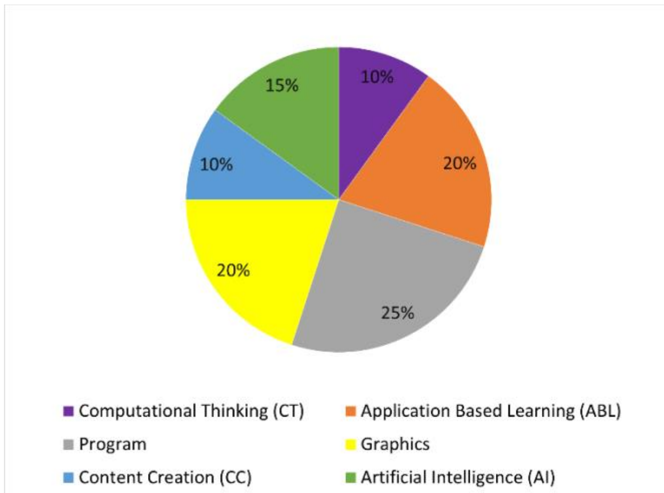

Figure 7. Flexible Curriculum

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).

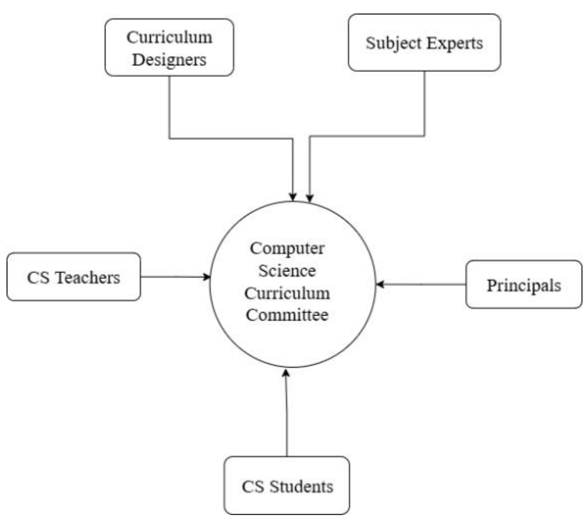

Figure 8. Structure of the CS Curriculum Development Committee

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