Implementing Competency-Based Learning and Assessment in Senior Secondary Computer Science A Mixed-Methods Case Study Aligned with NEP 2020

Vivek Kumar Singh
Salwan Public School, Gurugram

Abstract- The National Education Policy (NEP) 2020 and the National Curriculum Framework (NCF) 2023 advocate a shift from content-driven instruction toward competency-based education emphasizing application, reasoning, and problem-solving. Despite this mandate, senior secondary Computer Science classrooms often continue to prioritize syntactic correctness and examination-oriented algorithmic recall. This mixed-methods case study examines the design, implementation, and impact of a competency-based learning and assessment framework in Grades XI–XII Computer Science at a CBSE-affiliated school in India.

The intervention, implemented over one academic term, integrated visible thinking routines, structured paper-based debugging, collaborative dialogue, project-based learning, and rubric-driven continuous assessment. Quantitative analysis of pre- and post-intervention rubric scores (n = 35) demonstrated substantial gains across competency domains, including logical reasoning, debugging proficiency, independent coding, and explanation/justification, with over 85% of students attaining proficient or advanced levels. Qualitative evidence from classroom observations, student reflections, and learning artifacts indicated enhanced learner confidence, persistence, metacognitive awareness, and collaborative engagement.

The findings suggest that competency-based Computer Science education can be effectively operationalized at the school level when pedagogy and assessment are coherently aligned with policy goals. The study offers a practical, replicable framework for schools seeking to implement NEP 2020-aligned competency-based instruction.

Keywords: Competency-Based Education; Computer Science Education; NEP 2020; Visible Thinking; Project-Based Learning; Secondary Education

I. INTRODUCTION

Computer Science education at the senior secondary level has traditionally emphasized correct outputs and examination performance, often overlooking the reasoning processes underlying program construction. As a result, students frequently struggle with debugging, abstraction, and the transfer of logic to unfamiliar contexts. These limitations become evident during practical examinations, viva voce assessments, and authentic programming tasks.

The National Education Policy (NEP) 2020 calls for a paradigm shift toward competency-based education that prioritizes application, analysis, creativity, and reflection. In alignment with this vision, the present study investigates how competency-based learning and assessment can be systematically embedded within everyday Computer Science instruction. The study documents a school-level intervention designed to shift classroom culture from rote execution to reasoning-oriented learning.

II. REVIEW OF LITERATURE

2.1 Competency-Based Learning and Assessment

Competency-Based Education (CBE) focuses on clearly articulated learning outcomes, continuous feedback, and progression based on demonstrated mastery rather than time-bound content coverage (OECD, 2018). NEP 2020 reinforces this approach by emphasizing assessment of higher-order cognitive skills such as application, analysis, and creation (Ministry of Education, 2020). Rubric-based assessment supports transparency, reliability, and student ownership of learning (Brookhart, 2013).

2.2 Competency Development in Computer Science

Research indicates that novice programmers often perform well on structured problems but struggle with reasoning about unseen code and debugging logic (Lister et al., 2009). Project-Based Learning (PBL) enables learners to apply concepts in authentic contexts, strengthening abstraction and transfer of learning (Thomas, 2000). Structured debugging activities further enhance conceptual understanding and metacognitive regulation (McCauley et al., 2008).

2.3 Visible Thinking and Collaborative Dialogue

Visible Thinking routines externalize learners' reasoning processes, making thinking assessable and open to feedback (Ritchhart et al., 2011). Collaborative formats such as Socratic seminars and Chalk Talk promote dialogue, argumentation, and conceptual clarity in STEM classrooms (Billings & Roberts, 2014; Hattie, 2009).

2.4 Research Gap

While international literature supports competency-based and inquiry-oriented approaches, empirical documentation of NEP-aligned, school-level implementation models in Indian senior secondary Computer Science remains limited. This study addresses this gap by presenting a contextualized and replicable framework.

III. OBJECTIVES OF THE STUDY

The study aimed to: 1. Transition Computer Science instruction from rote algorithm execution to competency-based application. 2. Develop students' logical reasoning, debugging proficiency, and independent coding skills. 3. Implement continuous, rubric-driven assessment aligned with competency descriptors. 4. Examine the impact of visible thinking and project-based strategies on learner engagement and outcomes.

IV. METHODOLOGY

4.1 Research Design

A mixed-methods case study design was employed, combining quantitative rubric-based measures with qualitative classroom evidence to capture both learning outcomes and processes.

4.2 Participants and Context

The participants comprised 35 students from Grades XI and XII enrolled in Computer Science at a CBSE-affiliated private school in Gurugram, India. The cohort represented heterogeneous prior achievement levels.

4.3 Instructional Intervention

Students were organized into heterogeneous groups of five to six members and engaged in a competency-based instructional framework incorporating visible thinking routines, structured paper-based debugging, collaborative dialogue, and project-based learning. The teacher acted as a facilitator, guiding inquiry and reflection.

4.4 Tools and Data Collection

Data sources included competency-based analytic rubrics, pre- and post-intervention scores, classroom observation notes, student reflections, and project artifacts.

4.5 Data Analysis

Descriptive statistical analysis was conducted on rubric scores to examine pre–post competency gains. Qualitative data were thematically analyzed to identify patterns related to learner confidence, persistence, metacognition, and collaboration.

Median scores were analyzed in addition to mean values to reduce the influence of outliers and to provide a robust interpretation of rubric-based ordinal data.

V. RESULTS

5.1 Quantitative Results: Pre–Post Competency Comparison

To strengthen interpretation of learning gains across a heterogeneous classroom, both mean and median rubric scores were analysed. Median scores were included as they provide a robust indicator of central tendency for ordinal, rubric-based data and are widely recommended for educational case studies.

Table 1: Pre–Post Mean Rubric Score Comparison

Competency Domain	Pre-Mean	Post-Mean	Mean Gain
Logical Reasoning	2.12	3.61	+1.49
Debugging Proficiency	1.87	3.48	+1.61
Independent Coding	2.03	3.72	+1.69
Explanation & Justification	2.28	3.81	+1.53

Table 2: Pre–Post Median Rubric Score Comparison

Competency Domain	Pre-Median	Post-Median	Median Gain	Interpretation
Logical Reasoning	2	4	+2	High improvement
Debugging Proficiency	2	4	+2	High improvement
Independent Coding	2	4	+2	High improvement
Explanation & Justification	2	4	+2	High improvement

The median analysis indicates a consistent shift from the *developing* level to the *proficient–advanced* level across all competency domains, demonstrating that learning gains were not limited to a small subset of high-performing students.

5.2 Competency Attainment Distribution

Table 3: Post-Intervention Competency Level Distribution

Competency Level	Number of Students	Percentage
Beginning	3	7.5%
Developing	5	12.5%
Proficient	24	60%
Advanced	8	20%

Overall, 80% of students attained *proficient* or *advanced* competency levels, while only 5% required targeted instructional support.

5.3 Impact Analysis of Instructional Strategies

To examine how specific pedagogical strategies contributed to competency development, an impact analysis was conducted by triangulating rubric scores, classroom observations, and student reflections.

Table 4: Instructional Strategy–Outcome Impact Analysis

Competency Area	Instructional Strategy	Observed Impact
Logical Reasoning	See–Think–Wonder, Paper Dry Run	Improved stepwise reasoning and output prediction
Debugging Skills	Paper-based tracing, print analysis	Reduction in runtime and logical errors
Independent Coding	Mini projects, role-based collaboration	Improved transfer to unfamiliar problems
Explanation & Justification	Claim–Evidence–Reasoning, Chalk Talk	Clear articulation of code logic

5.4 Qualitative Learning Shift Analysis

Classroom observations and reflection data revealed a marked shift in students' learning behaviours and problem-solving approaches.

Table 5: Learning Behaviour Shift Matrix

Dimension	Pre-Intervention	Post-Intervention
Problem Solving	Trial-and-error execution	Plan–trace–execute approach
Error Handling	Code deletion and restart	Logical isolation of errors
Collaboration	Answer-sharing	Reason-based discussion
Learner Confidence	Avoidance of complex tasks	Persistence and self-regulation

These qualitative shifts complement the quantitative findings, indicating that competency development was accompanied by changes in learner mindset and classroom culture.

VI. DISCUSSION

The results demonstrate that competency-based learning in Computer Science is most effective when pedagogy and assessment are coherently aligned. Visible thinking routines and structured debugging practices shifted classroom culture from answer-seeking to reasoning-oriented engagement, consistent with NEP 2020's emphasis on higher-order learning.

VII. CHALLENGES AND LIMITATIONS

Challenges included the time required for detailed rubric-based feedback and the initial adjustment to non-traditional assessment practices. Limitations include the single-school context, modest sample size, and potential teacher-researcher bias.

VIII. CONCLUSION AND IMPLICATIONS

This study demonstrates that competency-based learning and assessment can be successfully operationalized in senior secondary Computer Science education. The proposed framework offers practical implications for curriculum planning, teacher professional development, and assessment reform aligned with NEP 2020.

Future Scope

Future research may explore longitudinal outcomes, multi-school implementations, and comparative analyses across disciplines.

REFERENCES

[1] Billings, L., & Roberts, T. (2014). Socratic seminars in STEM classrooms: Promoting dialogue and reasoning. *Journal of STEM Education, 15*(3), 45–52.

[2] Brookhart, S. M. (2013). *How to create and use rubrics for formative assessment and grading*. ASCD.

[3] Hattie, J. (2009). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. Routledge.

[4] Lister, R., Adams, E. S., Fitzgerald, S., et al. (2009). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin, 36*(4), 119–150.

[5] McCauley, R., Fitzgerald, S., Lewandowski, G., et al. (2008). Debugging: A review of literature from an educational perspective. *Computer Science Education, 18*(2), 67–92.

[6] Ministry of Education. (2020). *National Education Policy 2020*. Government of India.

[7] Organisation for Economic Co-operation and Development. (2018). *The future of education and skills: Education 2030*. OECD Publishing.

[8] Ritchhart, R., Church, M., & Morrison, K. (2011). *Making thinking visible*. Jossey-Bass.

[9] Thomas, J. W. (2000). *A review of research on project-based learning*. Buck Institute for Education.