THE IMPACT OF MATHEMATICAL CONCEPTS IN COMPUTER PROGRAMMING UNITS: A CASE STUDY OF THE KISUMU NATIONAL POLYTECHNIC

Felix Oduor and Silvance Abwao
The Kisumu National Polytechnic, Kenya

ABSTRACT

This study examined how mathematical proficiency shapes programming competence among trainees at The Kisumu National Polytechnic under the Competency-Based Education and Training (CBET) model. Using a mixed-methods design, the research assessed trainees' logical reasoning, problem-solving, and algorithmic thinking in relation to their understanding of algebra, Boolean logic, and discrete mathematics. Findings revealed a strong positive correlation (r = 0.78) between mathematical aptitude and programming performance. Trainees with higher mathematical proficiency showed better debugging, algorithm design, and perseverance. Weaknesses in math created barriers to mastering advanced fields like AI and cybersecurity. The study emphasizes integrating foundational mathematical concepts into programming courses to strengthen problem-solving and innovation skills in CBET institutions.

Keywords: Mathematical Proficiency, Programming Skills, CBET, Logical Reasoning

1.0 INTRODUCTION

Mathematics serves as the cognitive backbone of programming by enabling logical structure and systematic problem-solving. Within TVET institutions, the CBET model prioritizes hands-on instruction, often at the expense of theoretical rigor. This study investigates how foundational mathematics—discrete math, algebra, and Boolean logic—affects trainees' programming performance at The Kisumu National Polytechnic. It further explores whether mathematical gaps impede progress in advanced programming areas such as AI and cybersecurity. Insights from this research aim to inform curriculum improvements that better prepare trainees for the digital economy.

2.0 LITERATURE REVIEW

Mathematics and Programming Foundations Prior studies (Beaubouef, 2002; García-Santillán et al., 2015) assert that numeracy enhances computational logic and code efficiency. Prince (2023) identifies algebra, discrete math, and Boolean logic as vital for mastering data structures and algorithm design.

CBET and Theoretical Gaps

CBET focuses on practical competencies but may underrepresent essential theoretical components. Shim et al. (2017) note that insufficient math in CBET curricula hinders engagement with complex fields like AI and system optimization.

Computational Thinking and Logic

Wing (2006) and Guzdial (2019) argue for integrating mathematical logic into programming to cultivate computational thinking, a core skill in software development.

Performance and Persistence

Studies (Vihavainen et al., 2013; Liao & Bright, 1991) find that students with strong math backgrounds outperform peers in logic-based programming tasks. Perseverance and attention—key to coding success—are tied to structured mathematical learning.

Contextual Gap in TVET

Most literature focuses on university-level education. This study addresses the gap in TVET, specifically within CBET, where hands-on learning may dilute the mathematical rigor essential for programming mastery.

3.0 RESEARCH DESIGN

A mixed-methods design was employed, combining quantitative aptitude testing and programming tasks with qualitative interviews and surveys. A comparative framework assessed performance across three math proficiency levels.

Study Area

The Kisumu National Polytechnic, located in Kisumu County, Kenya.

Study Population

Trainees enrolled in CBET Level 5 and 6 computing programs and trainers teaching programming and mathematical logic.

Sampling Procedure

Stratified sampling based on trainees' math proficiency levels.

Data Collection Methods

Pre-assessment tests, programming tasks, surveys, and structured interviews.

Data Analysis

MAT scores and programming outcomes were correlated using Pearson's r ($r = 0.78$, $p < 0.05$). Thematic analysis via NVivo uncovered patterns in confidence, perseverance, and concept mastery. Data triangulation across surveys, tests, and interviews improved reliability. Standardized rubrics evaluated programming task performance across algorithm design, debugging, and logic application.

4.0 FINDINGS AND DISCUSSIONS

High Math Proficiency (HMP) trainees consistently outperformed others in logic, debugging, and algorithmic design. A positive correlation ($r = 0.78$) confirmed math's role in programming performance. Boolean logic, recursion, and data structures posed greater challenges for Low Proficiency (LMP) trainees. HMP participants reported higher confidence and perseverance; LMP students often resorted to trial-and-error approaches. Trainer feedback affirmed that math-strong students grasped programming concepts more quickly. The CBET curriculum was found to underemphasize critical mathematical content. Despite challenges such as varied math backgrounds and short study duration, the findings strongly support integrating more mathematical instruction in CBET programming.

5.0 CONCLUSION

Mathematical foundations are instrumental in developing programming logic, efficiency, and persistence. Strong numeracy equips trainees to navigate complex computational tasks and prepares them for success in the evolving digital economy. Current CBET curricula must more fully integrate math for sustainable skill development.

6.0 RECOMMENDATIONS

- Integrate discrete mathematics and Boolean logic into all programming units.
- Offer remedial math training to support low-proficiency trainees.
- Align programming tasks with real-world mathematical applications.
- Standardize curriculum guidelines emphasizing math-programming linkages.

REFERENCES

1. Beaubouef, T. (2002). Why computer science students need math. ACM SIGCSE Bulletin, 34(4), 57-61.
2. García-Santillán, A. et al. (2015). Mathematical thinking and problem-solving. Int. J. Math. Ed., 46(8), 1085-1100.
3. Prince, R. (2023). Mathematical foundations for algorithmic thinking. Journal of Computational Pedagogy, 18(3), 204-218.
4. Shim, J. P. et al. (2017). Past, present, and future of decision support technology. Decision Support Systems, 33(2), 111-126.
5. Wing, J. M. (2006). Computational thinking. Comm. of the ACM.
6. Guzdial, M. (2019). Rethinking programming education. Journal of Computer Science Education.
7. Vihavainen, A. et al. (2013). Predicting performance in introductory programming courses. ACM Transactions on Computing Education.
8. Liao, Y. K. & Bright, G. (1991). Effects of mathematical logic instruction on programming achievement. Computers & Education.
9. UNESCO-UNEVOC (2022). AI in TVET systems.
10. ILO (2021). Skills for a Resilient Youth: Global Trends.