OPEN ACCESS

Manuscript ID:
EDU-2025-14019542

Volume: 14

Issue: 1

Month: December

Year: 2025

P-ISSN: 2320-2653

E-ISSN: 2582-1334

Received: 26.09.2025

Accepted: 01.11.2025

Published Online: 01.12.2025

Citation:
Cuniah, C., Panchoo, S., & Jaillet, A. (2025). Error-Grid Framework: A Pedagogical Approach for Diagnosing Programming Mistakes and Enhancing Learning Outcomes in Secondary Education. *Shanlax International Journal of Education, 14*(1), 49–58.

DOI:
https://doi.org/10.34293/education.v14i1.9542

Error-Grid Framework: A Pedagogical Approach for Diagnosing Programming Mistakes and Enhancing Learning Outcomes in Secondary Education

Canayah Cuniah
University of Technology, Mauritius
https://orcid.org/0000-0003-4764-3946

Shireen Panchoo
University of Technology, Mauritius

Alain Jaillet
University of Cergy-Pontoise, France

Abstract
Programming in secondary schools is still a difficulty because learners are transitioning to text-based languages; this is the case of programming. The dropping pass rates in Mauritius are indicative of endemic conceptual and procedural discontinuities. This study constructs and tests the Error-Grid Framework, a low-cost classroom intervention that uses errors as a kind of diagnostic message to inform specific instruction. Content analysis of 30 Grade 11 students (n=90 handwritten scripts; 2,987 lines) was conducted, providing a code of recurring errors in 11 concepts of core programming. Reliability was studied among four educators using Krippendorff's alpha, and effectiveness was studied with the help of a pre/post design with two groups of students (n=11; n=13). Classroom utility was measured using survey data from 39 teachers across 20 schools. The findings indicated moderate inter-rater reliability (1=0.67; 2=0.64), substantial error decreases (p<0.001), and continued difficulties in loops, arrays, and functions. Teachers showed dense usability and diagnostic worthiness in the study. The framework facilitates differentiated teaching and timely feedback. In the future, this work should be extended to other languages, where automation is introduced to provide scalability and investigate long-term effects.

Keywords: Programming Education, Error Analysis, Secondary Education, Diagnostic Tools, Teacher Support, Computational Thinking

Introduction

This is because programming has been a marketable skill in the current technology-oriented society to achieve a career in computing, automation, data processing, and problem-solving. At the secondary level, it is assumed that students acquire cognitive thinking skills and logical reasoning, although learning and teaching are difficult. Novices are forced to learn abstract concepts, bizarre grammar, and complex thinking, which often leads to frustration and ineffective performance (Izu & Mirolo, 2024). Programming exams have a declining pass rate in Mauritius, suggesting knowledge and teaching gaps. Other languages, such as Scratch and Alice, have made entry more accessible by simplifying syntax at the expense of seldom helping to eliminate conceptual mysticism that can occur when using text-based languages. Beginners have the most difficulty with compilers that identify syntax errors but not logical errors (Prather et al., 2020). Teachers also lack regularised ways of studying errors and organising particular interventions, which leads to an increase in the number of mistakes and the development of a lack of confidence (Demirdag, 2015). While more emphasis is placed on learning