

# When Lyrics Become Code: Programming Like a Rock Star

Ildar Akhmetov  
i.akhmetov@northeastern.edu  
Northeastern University  
Vancouver, BC, Canada

## Abstract

This workshop introduces educators to *Rockstar*,<sup>1</sup> an esoteric programming language that looks and feels like writing rock song lyrics. Although we do not advocate teaching entire CS1/CS2 courses with Rockstar, we believe there is room in the CS curriculum for esoteric languages as an *additional activity*, a *co-curricular workshop*, or part of a *student club*. By stepping outside the familiar realm of Python, Java, or C, participants learn to refocus on problem-solving and foundational logic rather than standard syntactic patterns. Moreover, because esoteric languages are relatively unknown to AI models, they can help reduce over-reliance on automated solutions. Attendees will learn the basics of Rockstar, reflect on how to integrate it into teaching contexts for enriched conceptual learning, and enjoy the playful fusion of code and rock songs.

## CCS Concepts

• **Social and professional topics** → **Computing education**.

## Keywords

esoteric programming languages, Rockstar, computing education, creative coding

## ACM Reference Format:

Ildar Akhmetov. 2025. When Lyrics Become Code: Programming Like a Rock Star. In *Proceedings of the 27th Western Canadian Conference on Computing Education (WCCCE)*. April 28–29, 2025, Calgary, Canada. 1 page. <https://doi.org/10.60770/M5J4-MQ15>

## 1 Workshop Title and Duration

**When Lyrics Become Code: Programming Like a Rock Star**  
**Expected Duration:** 1 hour 30 minutes

## 2 Presenter and Background

**Presenter:** Ildar Akhmetov is the director of computing programs for Vancouver and an associate teaching professor in the Khoury College of Computer Sciences at Northeastern University. He has taught at universities in Russia and Canada, with research interests in computing education and software engineering. Since 1998, he has learned at least one new programming language each year, spanning widely used options to esoteric ones like Rockstar.

<sup>1</sup><https://codewithrockstar.com/>



This work is licensed under a Creative Commons Attribution International 4.0 License.

## 3 Workshop Description and Format

This workshop is designed *for educators* seeking novel ways to enrich their teaching practice. We assume participants have basic proficiency in at least one programming language (e.g., Python, Java, C++) and have never encountered Rockstar before. Rather than substituting Rockstar for a standard CS1/CS2 curriculum, the goal is to present it as a valuable supplement—for workshops, clubs, or short classroom demonstrations. Participants will discover how the playful, lyrical nature of Rockstar helps emphasize core problem-solving skills while simultaneously reducing reliance on AI “auto-solutions.” Ultimately, the session aims to offer a fun, creative environment that celebrates the joy of programming and the spirit of rock music.

### 3.1 Learning Outcomes

By the end of this workshop, participants will: *Understand* Rockstar’s key syntactic elements (variables, poetic numbers, arithmetic constructs); *Experience* writing and executing Rockstar programs in a web-based environment; *Reflect* on and discuss how esoteric programming can enrich mainstream curricula as an extracurricular or teaching supplement; *Enjoy* the creative process of turning code into an audio “performance” via AI tools.

### 3.2 Session Outline

- (1) **Hello, Rockstar (5 min)**. Demonstration of a simple “Hello Vancouver!” in Rockstar, highlighting lyrical print statements.
- (2) **Introduction (5 min)**. Presenter background and overview of why esoteric languages, especially Rockstar, can promote deeper conceptual engagement.
- (3) **Rockstar Basics (20 min)**. Walkthrough of variables, poetic numbers, and arithmetic structures. Live coding demonstration to show how these features look and feel.
- (4) **AI Performance (5 min)**. Generating a short Rockstar program and letting an AI tool “sing” it, showcasing the playful synergy between code, lyrics, and tech.
- (5) **Small Group Experimentation (15 min)**. Educators form pairs or small groups, writing Rockstar programs for simple tasks (arithmetic, printing lines, etc.).
- (6) **Group Sharing & Discussion (10 min)**. Volunteers run and share their lyrical code; reflections on surprise and engagement.
- (7) **Further Example: Temperature Conversion (20 min)**. Collaboration on a more complex example using poetic numbers and descriptive variables.
- (8) **Pedagogical Opportunities (20 min)**. Discussion on how Rockstar can be integrated into the classroom or clubs as an additional resource. Brainstorming ways to reduce AI dependency and spark genuine enthusiasm for coding fundamentals.