CodeQuest

Project Proposal

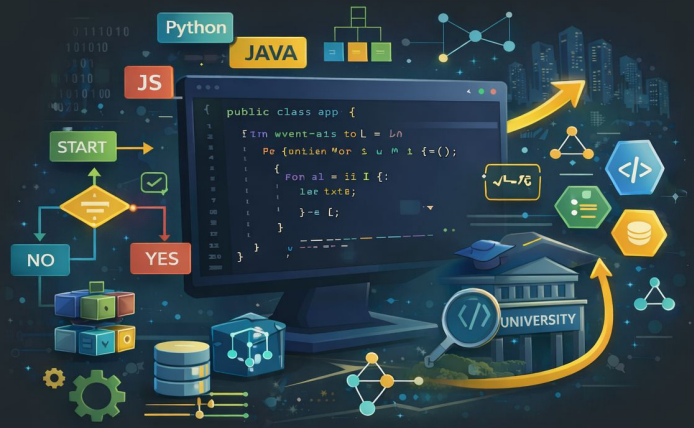
By: Anthony, Mikhail, Victor and Ajani

Background: The State of Coding Education

Computer Science in High School

Focuses on:

- Algorithms
- Abstraction
- Data structures
- Object-oriented design

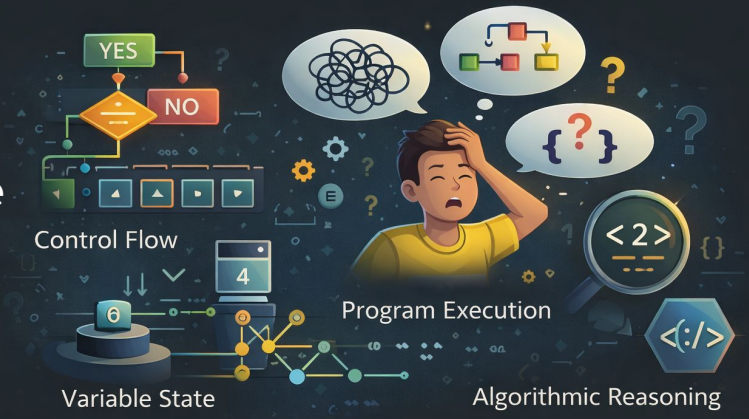


Background: The State of Coding Education

Research shows students can write syntactically correct code

But ***struggle*** with...

- Control flow reasoning
- Variable state tracking
- Understanding how programs execute
- Algorithmic thinking



Who’s our target audience?

Primary (core users):

- Grade 10–12 students aiming for **Computer Science / Engineering / STEM**
- Students in **AP Computer Science A** or **provincial CS courses**
- Motivated **self-directed learners** preparing for university

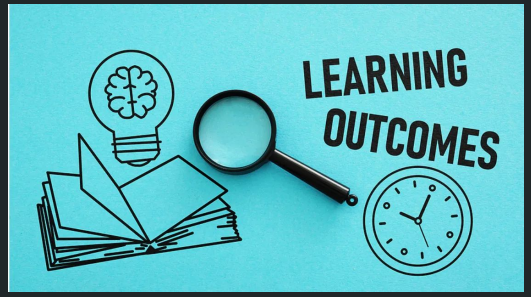
The knowledge gap we target:

- Interpreting **pseudocode** and reading unfamiliar solutions
- **Tracing program logic** (loops, conditionals, recursion) and predicting output
- Translating steps into working code + explaining **why** it behaves that way

Secondary audiences (also benefits):

- Students who feel behind and need **structured practice** to catch up
- **Teachers/tutors** using it for reinforcement, targeted review, and progress tracking

Learning Outcomes



What Students Will Be Able To Do

Measured through: pseudocode interpretation • tracing • implementation • explanation

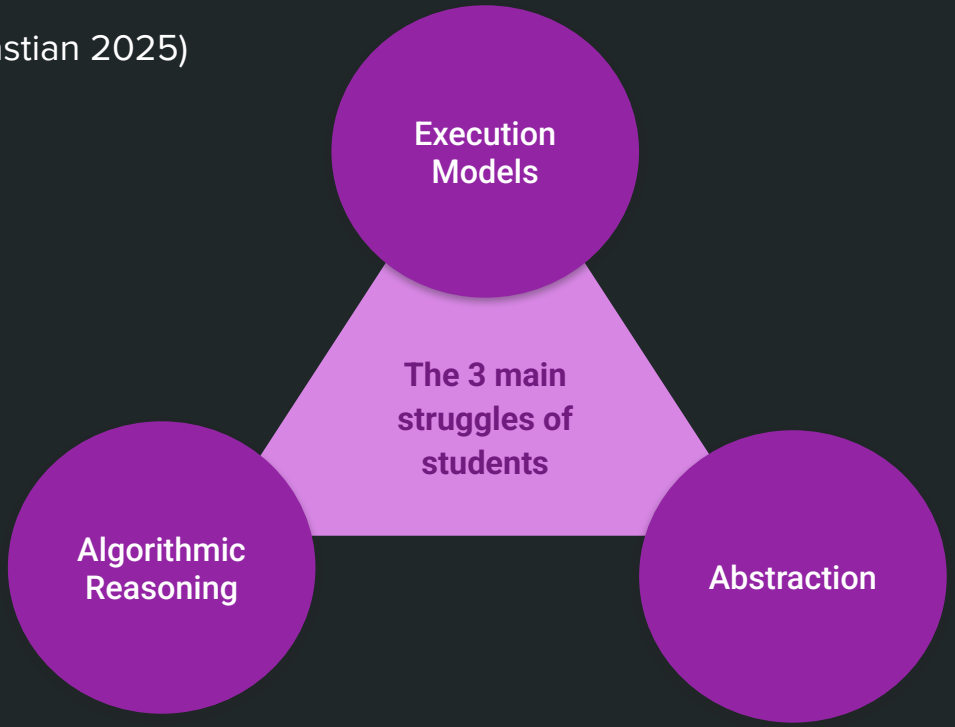
- **Algorithmic Reasoning:** compare solutions + explain efficiency using **Big-O**
- **Data Structures:** build/select structures (arrays, lists, stacks/queues, trees, hash) based on **trade-offs**
- **Execution Model:** trace control flow + variables; understand **recursion** and the **call stack**
- **Object-Oriented Design:** apply **abstraction/encapsulation**; design clean classes and responsibilities
- **Debugging + Explanation:** find logical errors and explain **why** code behaves the way it does
- **Collaborative Practices:** use basic **version control workflows** (commits, branches, PRs)

Related Work: Current Programming Learning Platforms

Platform	Strength	Limitation
Scratch	Great for beginners	No formal syntax or complexity
Codecademy	Interactive syntax tutorials	Linear, limited complexity focus
LeetCode	Strong algorithm practice	Assumes prior mastery

What Research Says Students Struggle With

- Persistent misconceptions in control flow (Bastian 2025)
- Logical errors reflect deeper misunderstanding (HoQ et al. 2025)
- Students struggle to articulate structural reasoning (McCracken et al. 2024)
- Educators report difficulty teaching algorithms & data structures (Malmi 2025)



How Code Quest Responds to Research

Research Gap	Code Quest Feature
Execution misconceptions	Syntax Sprint
Weak Big-O reasoning	Big-O Challenge Arena
Abstraction difficulties	Data Structure Builder
Recursive confusion	Recursion & Memory Lab
Poor code explanation skills	Debugging Quests
Lack of workflow exposure	Git Quest

Bridging High School and University CS

High School

- Basic Syntax
- Guided Exercises
- Surface Exposure

University

- Big-O analysis
- Data Structure trade-offs
- Formal Abstraction
- Independent Problem Solving