C-ing Beyond AI: A Cloud-Based Approach to Authentic Programming Assessment

Ildar Akhmetov
Khoury College of Computer Sciences
Northeastern University
Vancouver, B.C., Canada
i.akhmetov@northeastern.edu

Maryam Tanha
Khoury College of Computer Sciences
Northeastern University
Vancouver, B.C., Canada
m.tanha@northeastern.edu

Logan W. Schmidt
Khoury College of Computer Sciences
Northeastern University
Vancouver, B.C., Canada
l.schmidt@northeastern.edu

Saeed Yazdanian
Khoury College of Computer Sciences
Northeastern University
Vancouver, B.C., Canada
s.yazdanian@northeastern.edu

Abstract

Recent advances in large language models (LLMs) introduce unprecedented challenges for academic integrity in programming courses. This paper presents a cloud-based programming assessment system that creates ephemeral coding environments to preserve the authenticity of student work and deter AI-assisted plagiarism. Using Terraform and AWS, the system provisions individualized virtual machines for in-person assessment, mirroring the course environment without granting access to pre-existing code or external resources. Integrated with GitHub Classroom, the system handles assignment distribution, code submission, and resource clean-up. We discuss the design, cost analysis, and preliminary observations from implementation in a CS 2 course in C at Northeastern University (Vancouver). Preliminary results indicate that this controlled environment promotes student engagement and discourages reliance on AI for routine tasks. Future work will include studying how this approach impacts learning outcomes and AI usage patterns.

Keywords

Assessment, Programming, Cloud Computing, Academic Integrity, Large Language Models, Terraform, AWS

ACM Reference Format:
Ildar Akhmetov, Maryam Tanha, Logan W. Schmidt, and Saeed Yazdanian. 2025. C-ing Beyond AI: A Cloud-Based Approach to Authentic Programming Assessment. In *Proceedings of the 27th Western Canadian Conference on Computing Education (WCCCE). April 28–29, 2025, Calgary, Canada.* 2 pages. https://doi.org/10.1145/10.60770/4bhf-g950

1 Introduction

Recent developments in large language models (LLMs) such as ChatGPT, Gemini, and Claude pose new risks for academic integrity. Students can easily use AI tools to solve homework or take-home exams, making it difficult to assess their mastery of fundamental concepts. While remote-proctoring solutions and plagiarism detection tools have been used to mitigate cheating, these methods often fail to match the pace of AI advancements and raise ethical and privacy concerns [1, 2, 5, 8, 9].

Educational researchers have examined how LLMs transform both teaching and learning, highlighting the need for new instructional designs and assessment strategies that respond to AI technology's rapid diffusion [4, 7]. As evidence shows that reliance on AI for routine tasks can hinder deeper conceptual understanding, it is imperative to redesign assessments that accurately measure knowledge acquisition. Educators have proposed incorporating in-person practical evaluations, code-walks, and assignments that emphasize problem-solving and critical thinking over simple solution generation.

In line with the growing trend of online evaluation tools [3, 6], this paper describes a cloud-based approach to creating short-lived, controlled coding environments. Our goal is to preserve student work authenticity by mirroring the course environment while preventing access to previously saved code or external resources. We implemented this system in a CS2 course (algorithms, data structures and basic computer systems using C) at Northeastern University (Vancouver). While not a formal study, preliminary observations suggest that controlled, face-to-face assessments may reduce reliance on AI assistants and encourage deeper engagement with course material. This paper outlines the system design, cost-effectiveness, and avenues for future research.

2 Cloud-based Programming Assessment System

Our cloud-based assessment system provides ephemeral, isolated programming environments that replicate the environment used in the course. The course requires students to write C programs using a specific toolchain on Rocky Linux, employing vim as the primary text editor. Our goals are summarized as follows: *Faithfully emulate* the course environment for in-person exams; *Prevent* students from accessing pre-existing work or external resources; *Allow* instructors to manage student instances efficiently; *Minimize* complexity and cost.

WCCCE '25, Calgary, AB, Canada

https://doi.org/10.1145/10.60770/4bhf-g950

2.1 System Architecture

We use Terraform[1] for infrastructure as code, deploying individual AWS EC2 instances for each student at the beginning of the assessment. These instances are destroyed upon completion. GitHub Classroom serves as the assignment distribution and submission platform, integrated through the GitHub REST API to automate repository creation and access control.

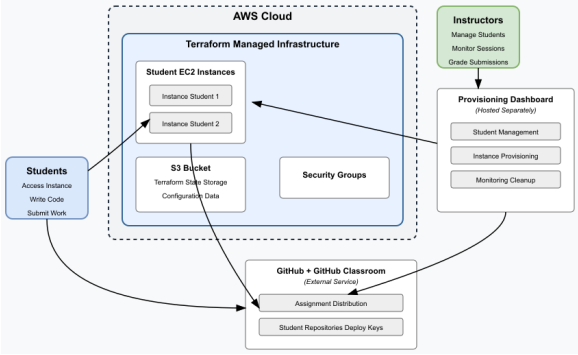

Figure 1: Architecture of the cloud-based programming assessment provisioning system

2.2 Implementation Details

Our Python Flask application offers a dashboard that manages student lists, provisions EC2 instances, and monitors their lifecycle. It imports GitHub usernames for upcoming sessions, generates and configures EC2 instances using Terraform, assigns each instance a unique password and GitHub deploy key for repository access, and tracks the instance states. Once assessments have been completed, the application displays instance details and initiates their destruction to clean up resources.

Each instance is configured with Rocky Linux, gcc/clang, vim, and any additional dependencies required. It clones the student's repository from GitHub upon launch and allows the student push their changes via the deploy key. Once the student's session ends, instructors use the dashboard to terminate the instance and revoke write access to the repository (using GitHub REST API).

2.3 Cost Analysis

We evaluated system costs during the Spring 2025 offering of CS2, which enrolled 71 students. Students were assessed in groups of 9 and worked for up to one hour on a `t2.small` AWS instance. Over multiple sessions, the total AWS cost was approximately CA$3.00, or CA$0.04 per student. To control costs, the system provisions instances only for upcoming sessions (usually 1–2 hours in advance) and promptly tears them down after completion.

3 Future Work

Our observations suggest students increased their engagement with course materials following the in-person coding assessment. Several reported a shift away from routine reliance on AI tools, recognizing

[1]https://www.terraform.io

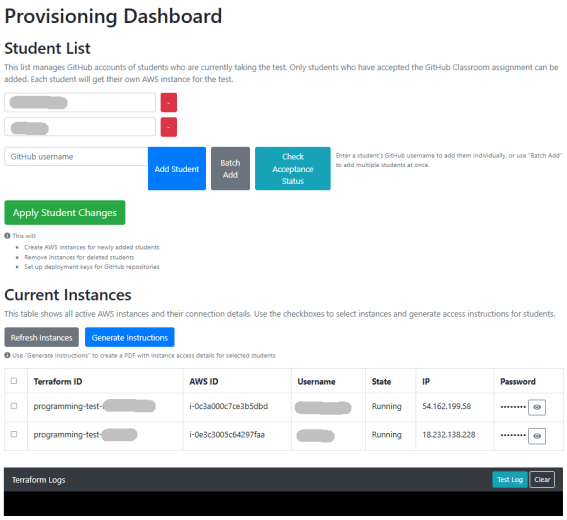

Figure 2: Screenshot of the provisioning dashboard showing instances for each student

that authentic practice was essential for success and mastery of fundamental course concepts. In upcoming semesters, we plan to conduct a study on the system's impact on learning outcomes and AI usage, expand its use in other programming-intensive courses, and release the source code and documentation for broader educational adoption. By exploring these directions, we will refine methods for assessing genuine coding ability in an era where AI assistance is increasingly prevalent.

References

[1] Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2024. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in education and teaching international* 61, 2 (2024), 228–239.

[2] Héctor Galindo-Domínguez, Lucía Campo, Nahia Delgado, and Martín Sainz de la Maza. 2025. Relationship between the use of ChatGPT for academic purposes and plagiarism: the influence of student-related variables on cheating behavior. *Interactive Learning Environments* (2025), 1–15.

[3] J Geetha, DS Jayalakshmi, E Naresh, and N Sreenivasa. 2024. Lightweight Cloud-Based Solution for Digital Education and Assessment. *Science & Technology Libraries* 43, 3 (2024), 274–286.

[4] Ahmed M Hasanein and Abu Elnasr E Sobaih. 2023. Drivers and consequences of ChatGPT use in higher education: Key stakeholder perspectives. *European journal of investigation in health, psychology and education* 13, 11 (2023), 2599–2614.

[5] Muntasir Hoq, Yang Shi, Juho Leinonen, Damilola Babalola, Collin Lynch, Thomas Price, and Bita Akram. 2024. Detecting ChatGPT-generated code submissions in a CS1 course using machine learning models. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 526–532.

[6] Magendran Munisamy, SZ Osman, and Mageswaran Sanmugam. 2024. Code, click, learn: a systematic review of online assessment tools in 21st century programming education. *Int J Mod Educ* 6, 20 (2024), 358–377.

[7] Iris Cristina Peláez-Sánchez, Davis Velarde-Camaqui, and Leonardo David Glasserman-Morales. 2024. The impact of large language models on higher education: exploring the connection between AI and Education 4.0. In *Frontiers in Education*, Vol. 9. Frontiers Media SA, 1392091.

[8] Zachary Taylor, Cy Blair, Ethan Glenn, and Thomas Ryan Devine. 2023. Plagiarism in entry-level computer science courses using ChatGPT. In *Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*. IEEE, 1135–1139.

[9] Yunkai Xiao, Soumyadeep Chatterjee, and Edward Gehringer. 2022. A new era of plagiarism the danger of cheating using AI. In *20th International Conference on Information Technology Based Higher Education and Training (ITHET)*. IEEE, 1–6.