

# MOCSIDE: An Open-source and Scalable Online IDE and Auto-Grader for Computer Science Education

Jonathan Cazalas  
jcazalas@flsouthern.edu  
Florida Southern College  
Lakeland, FL, USA

Max Barlow  
mbarlow@flsouthern.edu  
Florida Southern College  
Lakeland, FL, USA

Ibraheem Cazalas  
icazalas@flsouthern.edu  
Florida Southern College  
Lakeland, FL, USA

Chase Robinson  
srobinson6@flsouthern.edu  
Florida Southern College  
Lakeland, FL, USA

## ABSTRACT

Programming is learned through practice, with said practice in introductory programming courses often translating to a prohibitively large number of assignments, increasing the grading workload for faculty and/or teaching assistants. In short, this is unsustainable. Several publishers and a few notable companies have provided meritable auto-grading solutions, although most are plagued with problems including minimal problem sets, limited customization options, high cost, and at times even a disconnect with the pedagogical needs of academia. This poster presents our newly-developed web application, MOCSIDE, an open-source and scalable online IDE and auto-grader for computer science education. Results indicate a positive user experience from students and instructors alike, with cost savings, ease of use, and code collaboration highlighted as key features.

## CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; • **Software and its engineering** → **Integrated and visual development environments**.

## KEYWORDS

auto-grading, online IDE, computer science education, CS1, CS2

## 1 INTRODUCTION

Born out of necessity, auto-grading systems began arriving on the scene as far back as 2007 and have evolved from primitive solutions merely testing against provided test cases or unit tests to an analysis of the actual code itself [2] and most recently even research into the feedback provided, arguing the need for detailed, formative feedback.[1] Much of the research in auto-grading assumes a one-size-fits-all approach, with little discussion on the size and scope of the graded assignments. Auto-graders have proven immensely valuable when used with medium-sized problems; they provide little value, however, on much larger programming assignments, where the size and content scope is too unwieldy for automatic feedback. And what of very small problems, where the objective is quite granular? A multi-layered formative feedback classifier may be unnecessary.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9071-2/22/03.

<https://doi.org/10.1145/3478432.3499125>

## 2 METHODS AND RESULTS

Our proposal to meet this challenge is two-pronged. We have developed an open-source and scalable online IDE and auto-grader web application. We also propose partitioning assignments into two groups: larger, more meaningful assignments and problem sets comprised of very small, scaffolded problems, inherently designed with the auto-grading goal in mind and perfectly suited for the formative feedback needs of the learners. We are able to leverage auto-grading on the smaller problems while still assessing broader software engineering outcomes with the larger assignments. The value of our developed web application is that it's free to use, open source, and includes scaffolded problem sets in both Java and Python. Whereas many auto-grading solutions instruct students to download the problem specification, solve the program within an editor or IDE on their own computer, and then revisit the auto-grader to upload their solution, our open-source web application provides a full-service, self-contained experience for students and faculty, reminiscent of the more popular paid solutions, Repl.it and Coding Rooms. Students merely navigate to the given problem set, select a particular problem, code the solution within the provided web-based IDE, test their code, and then submit and await immediate feedback, all within the same browser tab.

## 3 IMPLICATIONS AND FUTURE WORK

The major contribution of this work is the development of scaffolded problem sets, in both Java and Python, and an open-source IDE and auto-grader to support them. While we are in year one of the study, initial results indicate a positive user experience from students and instructors alike, with ease of use, cost savings, and being open source highlighted as key features. We plan to improve the application by including code collaboration and gamification via live, anonymous scoreboards. A future publication will delve into the technical overview of the system and will provide a thorough analysis of the results of our ongoing study.

## REFERENCES

- [1] Georgiana Haldeman, Andrew Tjang, Monica Babeş-Vroman, Stephen Bartos, Jay Shah, Danielle Yucht, and Thu D. Nguyen. 2018. Providing Meaningful Feedback for Autograding of Programming Assignments. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 278–283. <https://doi.org/10.1145/3159450.3159502>
- [2] Emlyn Hegarty-Kelly and Dr Aidan Mooney. 2021. Analysis of an Automatic Grading System within First Year Computer Science Programming Modules. In *Computing Education Practice 2021* (Durham, United Kingdom) (CEP '21). Association for Computing Machinery, New York, NY, USA, 17–20. <https://doi.org/10.1145/3437914.3437973>