Auto-grading System

Make the life easier for students and instructors

George Zhang: pegasuszhang@g.ucla.edu

The Problem

For Instructor:

- Time consuming

- Repetitive

- Annoying

For Student:

- Slow

- Uncertainty

What can we do?

Current Solutions

- Give complete testing scripts to the students

  - Problematic

  - Student can hack the solution by reverse engineering from the testing scripts

- Commercial Solution

  - Example: Grade Scope

  - Expensive

  - Poor customizability

  - Made by UC Berkeley

Commercial Pricing

	Basic		Complete		
	SOLO $1/ student / course Upgrade Now	**TEAM** $3/ student / course Upgrade Now	**SOLO** $3/ student / course Upgrade Now	**TEAM** $5/ student / course Upgrade Now	**INSTITUTION** Custom Agreement Contact Sales
Functionality					
Collaborative Grading	-	✔	-	✔	✔
Unlimited Course Staff	-	✔	-	✔	✔
Assignment Statistics	✔	✔	✔	✔	✔
Regrade Requests	✔	✔	✔	✔	✔
Full Grade Export	✔	✔	✔	✔	✔
Late Submissions	✔	✔	✔	✔	✔
AI-Powered Grading	-	-	✔	✔	✔
Code Autograder Platform	-	-	✔	✔	✔
Support					
Basic Email Support	✔	✔	✔	✔	✔
Dedicated Support 4 business hr response time	-	-	✔	✔	✔

30 x 100 x 5 = 15000

Open-source Solution

Autolab

- Developed by a team of undergraduate students in Carnegie Mellon University.

- Started in 2010. CMU has been using it ever since.

- Open source under Apache License 2.0.

- Unlimited Customizability

Features

Autograding

Grade any assignment in any language using any software package. Instantly.

Feature

Rank	Nickname	Score
🏆 1	Batman	100
🏆 2	Alan Turing	98
🏆 3	2b ‖ ! 2b	95
4	ByteMachine	91

Scoreboards

Encourage healthy competition with a real-time rank ordered scoreboard.

Feature

Code Annotation

Augment the autograde with additional feedback directly on the student's code.

```
def foo(bar):

    bar += 5
        Avoid magic numbers  -5

    while (bar < 15213):

        bar = foo(bar+1)
```

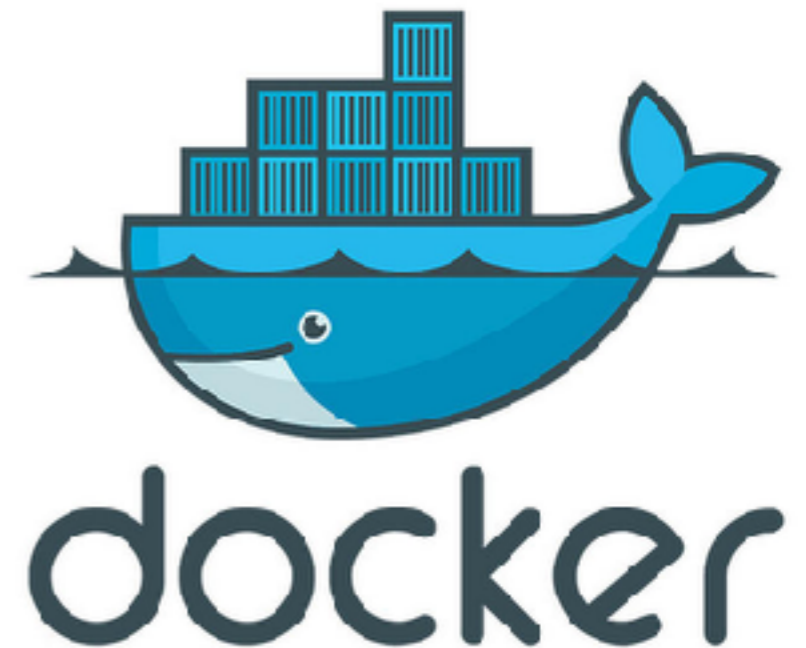# Feature


```
def foo(bar):
    bar += 1
    while (bar > 10):
        bar = foo(bar-2)
    if (bar == 15150):
        return bar - 1
    return bar + 1
```

```
def foo(bar):
    bar += 5
    while (bar < 10):
        bar = foo(bar+1)
    if (bar == 15112):
        return bar - 1
    return bar
```

Cheat Detection

Maintain academic integrity by comparing assessments with each other and with past submissions. Uses Stanford's Moss Cheat Detector.

Technology Stack

Interface for Instructors

- Mature and familiar open source toolchain

- Markdown for generating static homework instruction page

- Makefile for managing handout construction

- Write auto-grader script in any of your chosen language

- Grading result in JSON format

Experience for Students

- Simple one-click submission

- Specification checking

- Instant Feedback

- Rich Feedback Information

- Work until you get the grade you want

- No worry about missing something

What is left to do?

Challenge

- Primarily used inside Carnegie Mellon University. Deployed with the assistance of developers.

- Documentation is there but incomplete

- Main developers are busy seniors. No time to support us.

- Work with legacy code.

- Integration with UCLA IT Infrastructure.

Rewards

- Impactful

- Reachable

- Get full-stack development experience with cutting-edge technologies

- Agile Development

- Contribute to open-source project

Question?