

Kodethon: Tools for CS Education

CSU Sacramento
October 2017

UC DAVIS
UNIVERSITY OF CALIFORNIA

Martin Velez



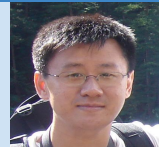
Michael Yen



Matt Le



Prof. Zhendong Su



A person's hands are shown typing on a laptop keyboard. The laptop screen displays lines of code, likely HTML or CSS, in a dark theme. To the right of the laptop, a white coffee cup sits on a saucer. The background is slightly blurred, focusing attention on the text and the person's hands.

Computer science is no more
about computers than
astronomy is about telescopes.

-- Source Disputed

What is **Computer Science?**

Turning **creative ideas** into **working systems**.

What is CS Education?

To **teach** students how to
turn **creative ideas** into **working systems**.

Teaching Experience

Students find programming **difficult**.

“How do I install Java?”

“How do I run this code?” (from Notepad)

Teaching programming is **difficult**.

“What programming projects should I assign?”

“How can I grade all submissions?”

“How can I detect plagiarism?”

The problem of scale:

Classes are large especially introductory classes with beginners.



“Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians.”

“The art of programming is the art of organizing complexity, of mastering multitude and avoiding its bastard chaos as effectively as possible.”



Edsger W. Dijkstra

Essential Complexity

Inherent and unavoidable issues of the problem being solved.

Write a function that *cubes* a number.

Write a compiler that translates a made up language E to C.

Write quicksort.

Complete programming assignment

Accidental Complexity

Issues related to the approach of the problem.

Do it in Python.

Do it in Java. Turn it on Friday.

Do it in Prolog. Work with a partner.

Go to CSIF basement to work with partner on a rainy night and find all the doors locked and your phone battery is dead.

Student Challenges

Setting up programming environments

Executing programs in different PLs

Collaborating with peers

Organizing files across devices

Other (e.g. Lack of interest)



How can we help **students** with these challenges?

Kodethon CDE: Write and Run Code Easily

An easy-to-use cloud development environment that helps students write and run code in many programming languages.

<http://www.kodethon.com>

Cloud Storage



Editor



\$ terminal



Features

- Zero Setup
- Magic Run Button
- Cloud Storage
- Real-time collaboration
- File History
- And more...

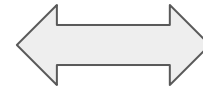
Programming In Java (Demo)

How do you build a cloud
development environment (CDE)?

Technical Challenges (File System)

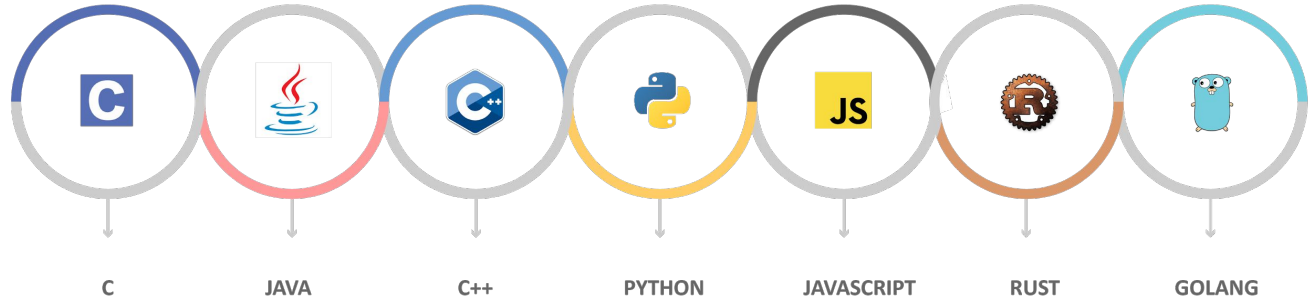
How do you ensure files are **readily available** to the user?

How do you ensure files are backed up?



Technical Challenges (General Purpose)

How do you support multiple, diverse programming environments?



Technical Challenges (Security and Scalability)

How do you support hundreds, thousands of concurrent users?



How do you ensure user files and operations are secure?

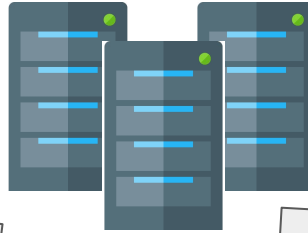


Kodethon's Architecture

User Interface



Master Nodes

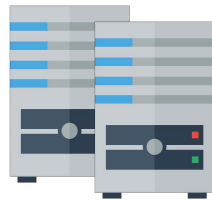


Database Server



Technologies

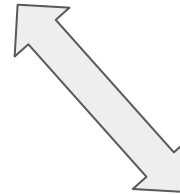
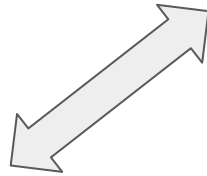
- Ruby-on-Rails
- AngularJS
- Docker
- PostgreSQL
- Firebase
- Google API
- *And more...*



File Servers



Compute Nodes



Feature Highlights (Demos)

Everything is a project!

Helps organize projects.

Facilitates sharing.

Real-time Collaboration

Facilitates pair programming

Enables Remote Assistance

CDE Shell

Easy-to-use

Sufficient for most needs

Unix Terminal

Transfer files

Full suite of shell commands

What about teachers?!

Some Teacher Challenges

Designing programming assignments

Grading and Feedback

Creating educational content

Tutoring

Other (e.g. Answering Piazza Questions)



UC Davis (400 students in ECS 30)

UC Berkeley (1762 students in 61A)

<http://www.dailycal.org/2017/08/24/introductory-computer-science-course-enrollment-increases-last-year/>

Kodethon Courses: Manage Homeworks Easily

An easy-to-use learning management system that helps teachers with the burden of grading and feedback.



The screenshot shows the Kodethon interface for a course demo. On the left, there is a sidebar with navigation options: Home (Demo #67), Assignments, Grades, Shared Files, and People. The main content area displays a table of assignments. The table has columns for Title, Start Date, and Due Date. One assignment is listed: 'Homework 1' with the description 'An introduction :)', a start date of '10/4/17 11:32 AM', and a due date of '10/20/17 11:32 AM'. Below the table, there is a 'Grade Assignment' button.

Title	Start Date	Due Date
Homework 1 An introduction :)	10/4/17 11:32 AM	10/20/17 11:32 AM

[Grade Assignment](#)



Assignments

Create, Release, Reuse



Auto Grading

Simple and Custom



Auto Test Case Generation

E.G., ECS10 HW, C° to F°

Kodethon Courses (Demo)

Ongoing and Future Work

CDE

- Metacompiler - more helpful compiler error messages
- Code Search - search for common snippets
- Visual Debugger - standard feature in IDEs

Courses

- Plagiarism Detection (e.g. MOSS)
- Improved Feedback
- Automatic problem generation (project name: Kodejoy)

Feature Highlight:
Embed
(Demo)



Martin Velez



Michael Yen

Kodethon



Matt Le



Prof. Zhendong Su

Computer Science is no more about computers than astronomy is about telescopes.

-- *Source Disputed*

<https://kodethon.com>

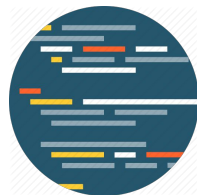
Store

- Cloud
- Up/Download
- Share



Edit

- Syntax highlighting
- Autocompletion
- Vim mode



Execute

- Java, Lisp, Prolog
- ssh
- git



Kodethon

<https://kodethon.com>

Leveraging the research expertise @ **UC Davis** to address the challenges of CS education

LMS - [Learning Management System](#)

ECS140A

HW2



Share

HW3



Share

ECS30

HW1



Share

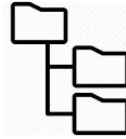
Checkout

Grade

CDE - [Cloud Development Environment](#)

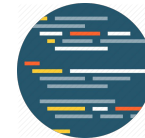
Files

- Cloud
- Up/Download
- Share



Editor

- syntax highlighting
- autocompletion
- vim mode



```
$ javac HW.java
$ java HW
- C/C++, gdb
- Python, Lisp
- Prolog
- ...
```



Assignments

Create, Release, Reuse



Auto Grading

Simple and Custom



Auto Test Case Generation

E.G., ECS10 HW, C° to F°

Demos

<https://www.youtube.com/channel/UC2sFV2b9TES3Dbo250oDXpg>

ECS140A HW2

<https://www.youtube.com/watch?v=kv7Al3FMMkE>

Auto Grading: Student View

<https://youtu.be/t72TNrKCZu4>



CDE Shell

More helpful compiler/interpreter error messages
Links to web help and examples



Community

Social sharing and communication



File History

Automatic backup of revision



LMS

Instant Feedback and Grading (via LMS)



Shared Files

Realtime online collaboration (a la "Google Docs")

Pair programming or TA Online Help

<https://www.youtube.com/watch?v=q4uZpuEIUYA>

<https://www.kodethon.com/#/courses>

<https://kodethon.com>