

# moodlemoot<sup>TM</sup>

PHILIPPINES 2024

***AI in Moodle: Transformative Teaching Technologies***

# CS Teacher VPL Tricks

*Half your time checking student programs while doubling student engagement with Moodle VPL*

---

Presented By:

Roy Vincent L. Canseco, MSEE, PhD Cand. CS  
Computer Science Unit / MIS  
Philippine Science High School - Main Campus



# Teacher and I.T. Practitioner

I have had the privilege of teaching at various institutions, including the Institute of Electrical and Electronics Engineering (IEEE), the Department of Computer Science (DCS) and the Virata School of Business at the University of the Philippines Diliman (UPD). During my time at UPD, I actively contributed to the development of UPD's Moodle-based UVLe LMS as a Project Development Associate.

Currently, I am teaching at Philippine Science High School - Main Campus, where I served as the head of the Information Technology Unit (MIS) during the deployment of our Moodle-based Learning Management System called KHub.



Roy Canseco

CS Teacher , MIS - PSHS-MC

# Talk Description

Join me as I share how our campus set up and leverages Moodle's Virtual Programming Laboratory (VPL), along with generative AI to heighten student engagement, while minimizing code checking time.

Discover how we have implemented these technologies to enhance the learning experience, while keeping server costs cheap and maintenance easy.

# CS TEACHER VPL TRICKS

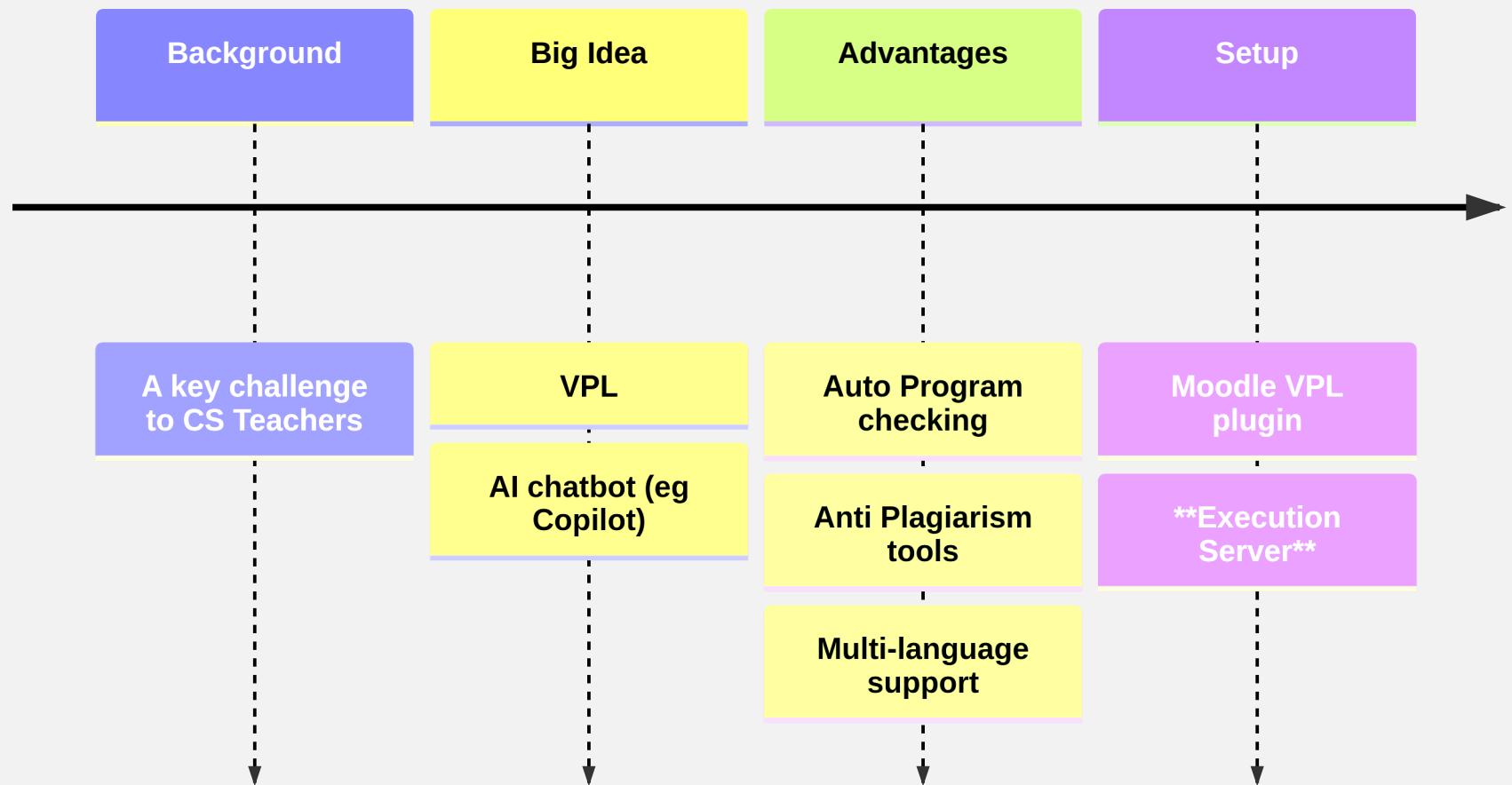
MoodleMoot 2024

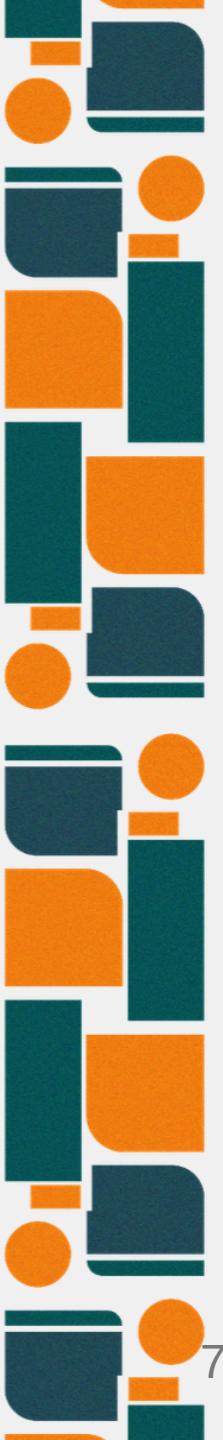
May 2024

Roy Vincent L. Canseco



# Talk Outline





According to a study funded by Google, under the direction of ACM, and in partnership with Microsoft:

"Computer science educators often struggle with limited access to **up-to-date materials, software, and hardware**"

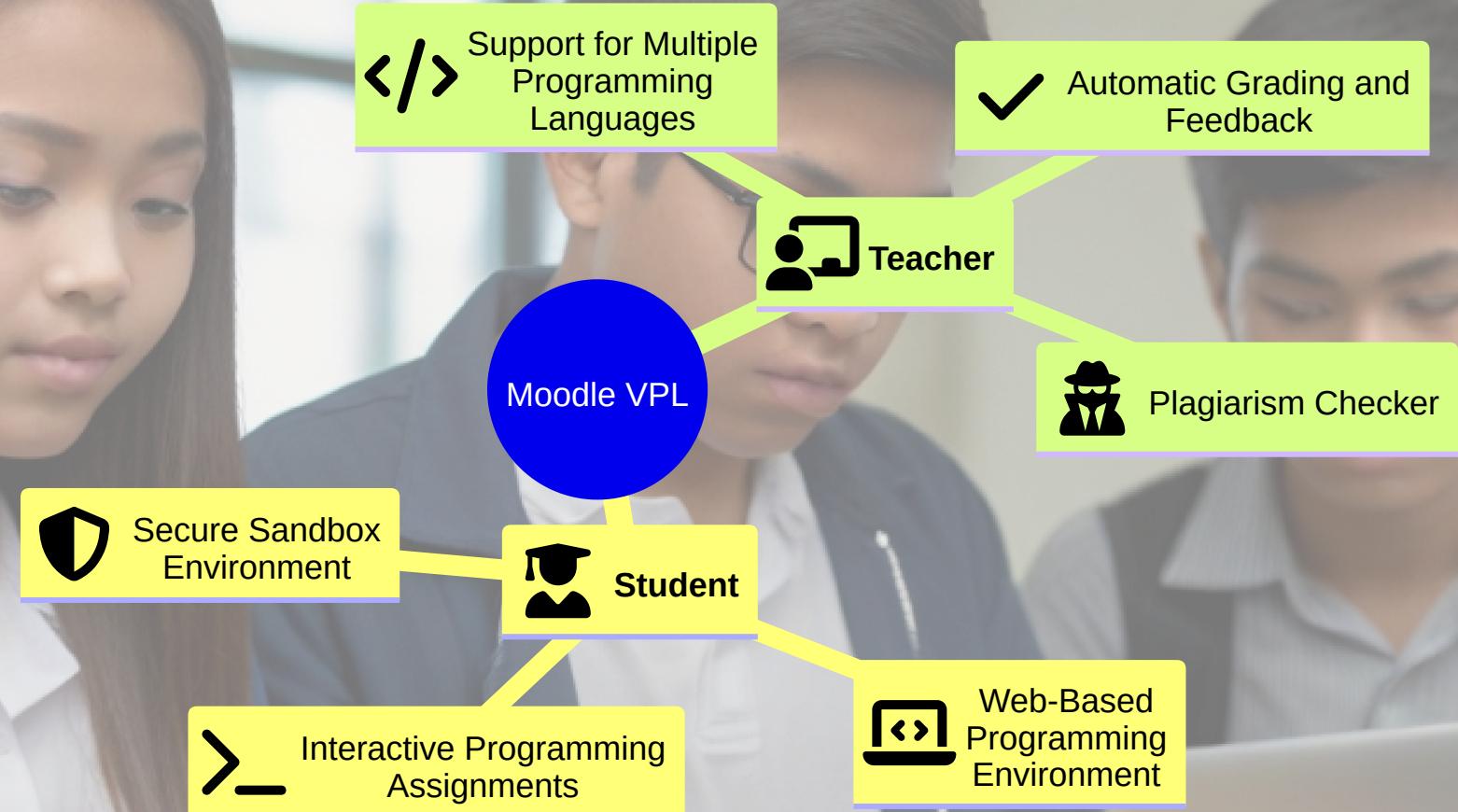
-University of Chicago Study

01

# VPL - Virtual Programming Laboratory

# Benefits

## Virtual Programming Laboratory



# Interactive Secure Web-based Environment

SY23.CS3.Web Development Q3 - PracTest - GN (Hard) Edit — Mozilla Firefox

index.js

```
1 process.stdout.write('Enter a number: ');
2
3
4 let input = [];
5 process.stdin.on('data', function (data) {
6   input = data.toString().trim();
7
8   let total = 0;
9
10  if (input.length >= 2){
11    arr = input.split(')').map(Number);
12
13    for (let i = 0; i < input.length; i++) {
14      total = Number(total) + Number(input[i])
15    }
16
17  }
18  else {
19    total = Number(input);
20  }
21
22  console.log(total);
23  process.exit();
24
25});
```

**Evaluate**

Evaluating

Proposed grade: 6.38 / 17

Comments

Failed tests

- Test 3: demo3
- Test 4: demo4
- Test 5: 1210
- Test 6: first five
- Test 8: love love

**Test 3: demo3**

Incorrect program output

--- Input ---  
47

--- Program output ---  
Enter a number: 11

--- Expected output (numbers)---  
2

Execution

Description

Ln 1, Col 1 JavaScript

# Automatic Code Similarity Checker

← → C https://khub.mc.pshs.edu.ph/mod/vpl/similarity/listsimilarity.php 120% ☆

Department of Science and Technology  
**PHILIPPINE SCIENCE HIGH SCHOOL SYSTEM**  
Scientific & Technological Service Institute

Home Dashboard More ▾ Edit mode

Virtual programming lab Settings Test cases Execution options Requested files More ▾

ID	File Name	Score / Total Score	Similarity Score	Contributors
78	index.js	8.50 / 17.00	62 91 70*	B E M il B ta (*)
				Yan en S ol (*)
79	index.js	4.25 / 17.00	64 84 48*	M I A a C z (*)
				BS B J ne S s (*)
80	index.js	8.50 / 17.00	50 91 59*	B E M il B ta (*)
				E D e u z (*)
81	script.js	17.00 / 17.00	54 87 72*	JL J F La lor (*)
				GN.js 17.00 / 17.00

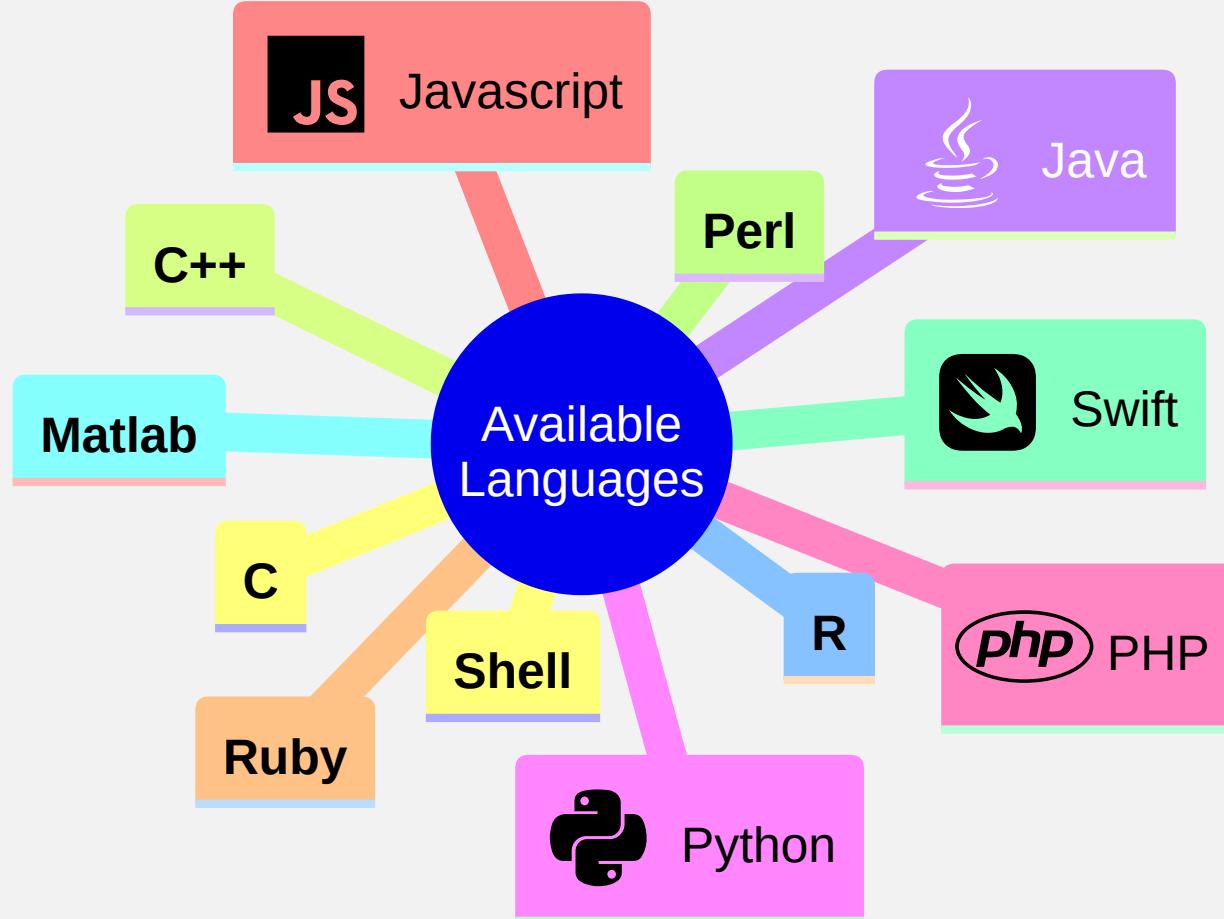
# Easy Code Comparison

diff — Mozilla Firefox  
https://khub.mc.pshs.edu.ph/mod/vpl/similarity/diff.php?type1=1&subid1=20048&filename1=script.js&type2=1&subid2=20273&filename2=GN.js 110% ⭐️

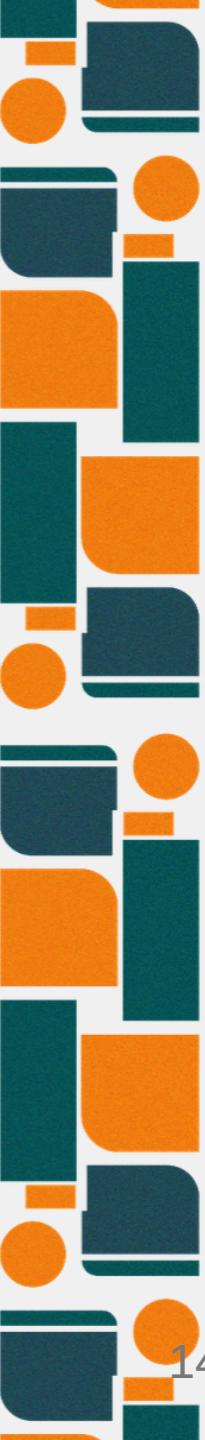
ipt.js	GN.js
1 process.stdout.write('Enter a number: ');\n2\n3 let input = '';\n4 process.stdin.on('data', function (data) {\n5     input = data.toString().trim();\n\n6     function sumOfDigits(num) {\n7         let sum = 0;\n8         while (num > 0) {\n9             sum += num % 10;\n0             num = Math.floor(num / 10);\n1         }\n2         return sum;\n3     }\n\n4     function g(n) {\n5         while (n >= 10) {\n6             n = sumOfDigits(n);\n7         }\n8         return n;\n9     }\n\n0     let result = g(parseInt(input));\n1\n2     console.log(result);\n3     process.exit();\n4 });	1 process.stdout.write('Enter a number: ');\n2\n3 let input = '';\n4 process.stdin.on('data', function (data) {\n5     input = data.toString().trim();\n\n6     function sumOfDigits(num) {\n\n7         let result = sumOfDigits(n);\n8         while (result >= 10) {\n9             result = sumOfDigits(result);\n10        }\n11        return result;\n12    }\n\n13    let result = g(parseInt(input));\n14\n15    console.log(result);\n16    process.exit();\n17 });

# Lots of Programming Languages supported

\*\* there are more  
depending on the setup



# Supported Programming Languages



Department of Science and Technology  
**PHILIPPINE SCIENCE HIGH SCHOOL SYSTEM**  
Scientific & Technological Service Institute

Home Dashboard My courses Site administration Course search

3 6 Edit mode

Virtual programming lab Settings Test cases **Execution options** Requested files More ▾

▼ Execution options

Based on	Select
Run script	JAVASCRIPT: Using nodejs with the first file
Debug script	FORTRAN: Using gfortran
Run	GO: Using "go build -o" with first file
Debug	GROOVY: Using default javac, run JUnit if detected
Evaluate	HASKELL: Using "ghc or runhugs +98" with the first file
Evaluate just on submission	HTML: Using "php -n -f" access index.html
Automatic grade	JAVA: Using default javac, run JUnit if detected
	JAVASCRIPT: Using nodejs with the first file
	JULIA: Julia running first jl file
	KOTLIN: Using default kotlinc
	LISP: Using clisp with the first file
	LUA: Using lua with the first file
	MATLAB: Using Matlab or Octave with the first file
	MINIZINC: Using "go run" with first file
	MIPS: MIPS R2000/R3000 using Spim emulator
	PASCAL: Using FPC or gpc
	PERL: Using "perl -w" with first file
	PHP: Using "php -n -f" with the first file or on serve if index.php exists
	PROLOG: Using swipl with the first file .pro or .pl
	PSEINT: Using pseint with the first file
	PYTHON-2: Using python2 with the first file

Contact us

Get the mobile app

VPL

02

# Recent Studies on A.I. for Computer Science Education



# Programming Is Hard – Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation

Brett A. Becker  
University College Dublin  
Dublin, Ireland  
[brett.becker@ucd.ie](mailto:brett.becker@ucd.ie)

Andrew Luxton-Reilly  
The University of Auckland  
Auckland, New Zealand  
[a.luxton-reilly@auckland.ac.nz](mailto:a.luxton-reilly@auckland.ac.nz)

Paul Denny  
The University of Auckland  
Auckland, New Zealand  
[paul@cs.auckland.ac.nz](mailto:paul@cs.auckland.ac.nz)

James Prather  
Abilene Christian University  
Abilene, Texas, USA  
[james.prather@acu.edu](mailto:james.prather@acu.edu)

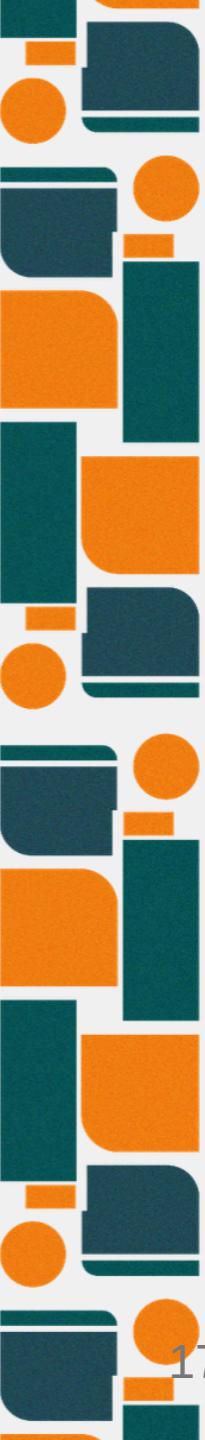
James Finnie-Ansley  
The University of Auckland  
Auckland, New Zealand  
[james.finnie-ansley@auckland.ac.nz](mailto:james.finnie-ansley@auckland.ac.nz)

Eddie Antonio Santos  
University College Dublin  
Dublin, Ireland  
[eddie.santos@ucdconnect.ie](mailto:eddie.santos@ucdconnect.ie)



© Canseco 2024

Figure 1: An image generated by Midjourney with the prompt “robot writing computer code while student watches, computer screens, computer programming, computer code, realistic, highly detailed, cinematic, aspect 16:9”



Becker et al, in 2023, discussed ways in which AI tools can help students learn programming:

1. **AI-generated solutions** can provide students with model answers to programming exercises
2. AI can generate **explanations of complex code**
3. Educators can leave low-level code to A.I. then **shift the focus to algorithms**
4. AI-generated starter code can help students

### **combat programmer's writer's block**

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming Is Hard – Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.

# Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming

Majeed Kazemitabaar

Department of Computer Science,  
University of Toronto  
Toronto, Ontario, Canada  
majeed@dgp.toronto.edu

Barbara J. Ericson

School of Information, University of Michigan  
Ann Arbor, Michigan, USA  
barbarer@umich.edu

Justin Chow

Department of Computer Science,  
University of Toronto  
Toronto, Ontario, Canada  
justinchow@dgp.toronto.edu

David Weintrop

College of Education, University of Maryland  
College Park, Maryland, USA  
weintrop@umd.edu

Carl Ka To Ma

Department of Computer Science,  
University of Toronto  
Toronto, Ontario, Canada  
cma@dgp.toronto.edu

Tovi Grossman

Department of Computer Science,  
University of Toronto  
Toronto, Ontario, Canada  
tovi@dgp.toronto.edu

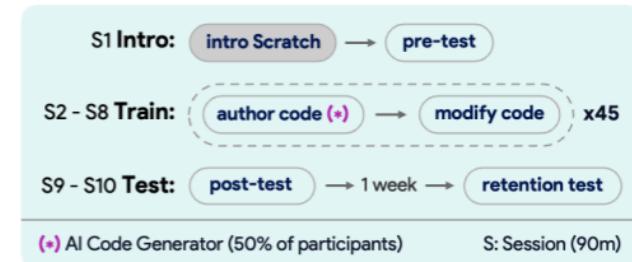
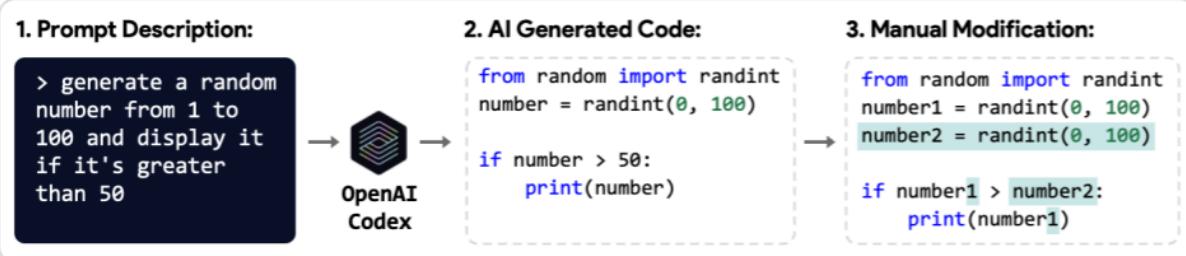
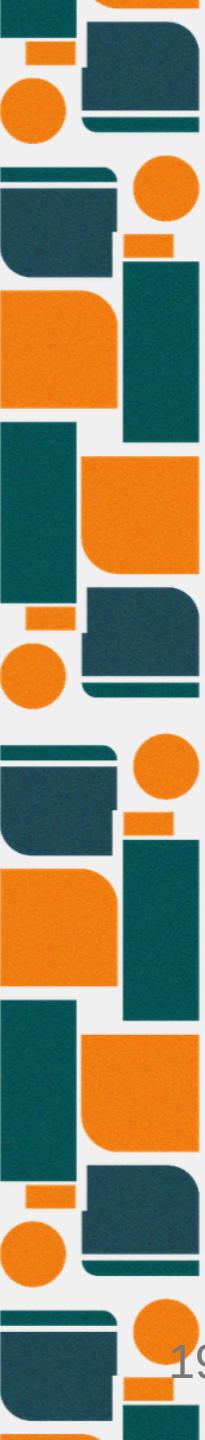


Figure 1: Left) Generate-modify usages with AI code generators. Right) Summary of our controlled study over 10 sessions.



In a 2023 study that involved 69 novices aged 10-17, Kazemitaab et al presented encouraging evidence that AI Coding Assistants can be integrated into programming education:

1. Students using AI code generators like Codex showed **completed tasks faster and with higher scores**,
2. Students with higher pre-test scores and access to Codex performed significantly **better in retention tests**
3. Students using AI code generators felt **less stressed and more eager** to continue learning programming
4. Students used AI code generators to **break tasks into subgoals** and solve each subgoal step by step

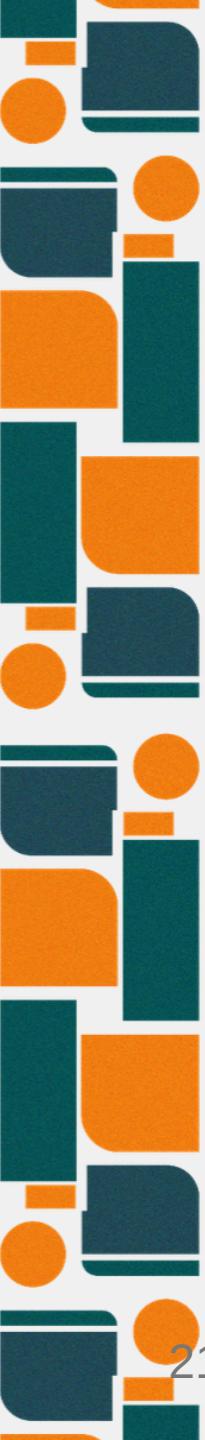
Kazemitaab, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). Studying the Effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany. <https://doi.org/10.1145/3544548.3580919>

***Proceeding Seminar Nasional & Call For Papers***  
ISSN Online: 2654-6590 | ISSN Cetak: 2654-5306

---

**ARTIFICIAL INTELLIGENCE (AI) IN EDUCATION: USING AI TOOLS FOR  
TEACHING AND LEARNING PROCESS**

**Tira Nur Fitria**  
Institut Teknologi Bisnis AAS Indonesia  
*Email:* [tiranurfitria@gmail.com](mailto:tiranurfitria@gmail.com)



Fitria, in 2021, highlights the following aspects of A.I. for education:

1. Students can have a more **Personalized Learning** experience with AI
2. A.I. will be **Complementing Teachers**
3. AI Applications can serve as **Virtual mentors / tutors**
4. A.I. carries potential technical issues like **cyberattacks and plagiarism.**

03

# VPL Secure and Cost-Effective Setup

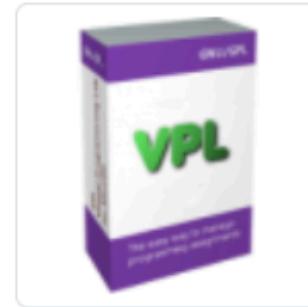


# VPL Setup and Use

1. ***Install VPL plugin*** - Installing Moodle VPL is a straightforward process that involves uploading the plugin package and configuring the settings.
2. ***Setup VPL Jail Server*** - Involves setting up a dedicated server, installing the necessary packages, configuring the server, and testing the installation.
3. ***Create VPL Assignments*** - Creating VPL assignments involves careful planning, designing, and testing to ensure they align with learning objectives and effectively assess programming skills.

## VPL Plugin

- Program in the browser.
- Run tests to grade the programs.
- Check for file similarity.
- Restrict copy-pasting.



# Virtual Programming Lab

Activities :: mod\_vpl



Maintained by Juan Carlos Rodríguez-del-Pino

Part of set [VPL - Virtual Programming lab](#).

VPL is an activity module to manage programming assignments

📍 Latest release: 3 months    ⚙️ 2166 sites    ⬇️ 2k downloads    ❤️ 156 fans

📄 Description

🏷️ Versions

📊 Stats

🌐 Translations

# VPL

Virtual Programming Lab for Moodle

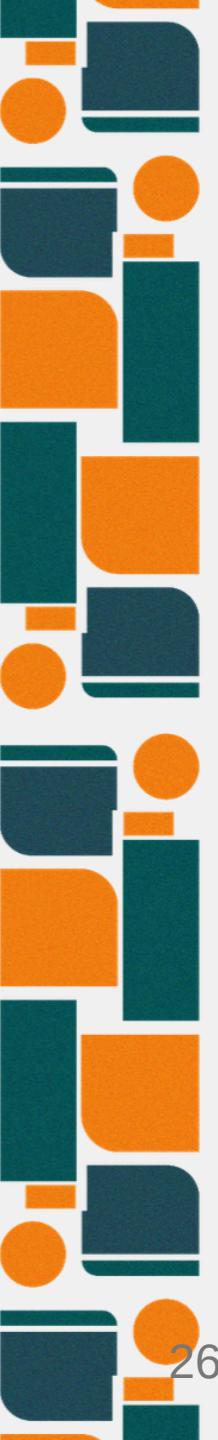
## Site Administration > Plugins > Activity Modules > Virtual Programming Lab > Execution servers config

--

### Execution servers list

[mod\\_vpl](#) | [jail\\_servers](#)

```
# This server is only for test use.  
# Install your own Jail server and remove the following line  
# http://demojail.dis.ulpgc.es  
http://[REDACTED]101.ip.linodeusercontent.com  
http://[REDACTED].101
```



# How did we get our own Jail server???

- VirtualBox servers (housed in our laptops)
- Local (LAN) servers here.
- Cloud servers! :)

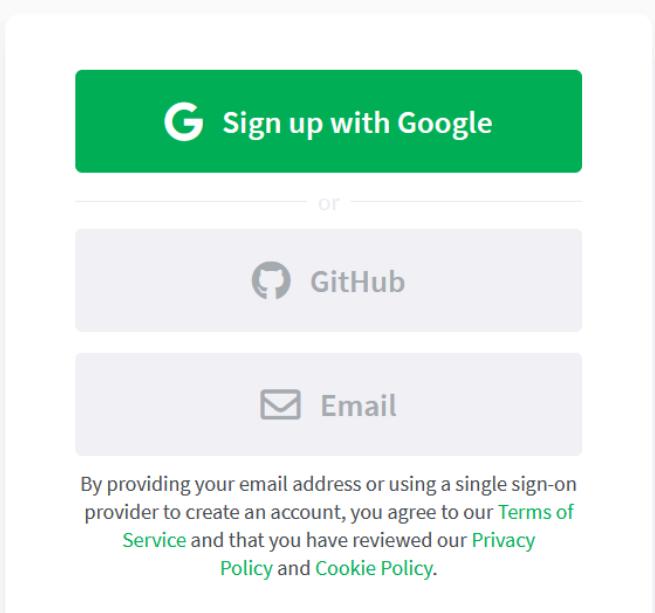
<https://www.linode.com>



# Cloud Computing Developers Trust

Build, run, and secure your cloud workloads on Akamai Connected Cloud, a massively distributed edge and cloud platform. [Sign up](#) today or [contact us](#) to learn more.

© Canseco 2024



By providing your email address or using a single sign-on provider to create an account, you agree to our [Terms of Service](#) and that you have reviewed our [Privacy Policy](#) and [Cookie Policy](#).

# 1 CPU, 2GB RAM, 50GB Storage

The screenshot shows the Linode Cloud interface. The left sidebar has a dark theme with white icons and text. The 'Linodes' item is highlighted with a green bar at the top. The main area displays a single Linode entry:

- Name:** vpl-moodle-backup
- Status:** RUNNING (green circle)
- Power Options:** Power Off, Reboot, Launch LISH Console, More...
- Summary:** 1 CPU Core, 50 GB Storage, 2 GB RAM, 0 Volumes
- Public IP Addresses:** (Redacted)
- Access:** SSH Access, LISH Console via SSH (Redacted)

The top navigation bar includes a search bar, a 'Create' button, and a user profile for 'pshsmc'. The bottom footer shows the MoodleMoot Philippines 2024 logo and the text '© Canseco 2024'.

# Ubuntu; Sg = \$12 USD / mo.

The screenshot shows the Linode cloud creation interface. On the left sidebar, under the 'Linodes' section, the 'Images' item is selected. In the main area, the 'Choose a Distribution' section has 'Ubuntu 20.04 LTS' selected in a dropdown menu, which is highlighted with a red box. Below it, the 'Region' section shows 'Singapore, SG (ap-south)' selected in a dropdown menu, also highlighted with a red box. The 'Linode Plan' section shows the 'Shared CPU' tab selected, with a note that Shared CPU instances are good for medium-duty workloads. A table lists three plan options: Nanode 1 GB, Linode 2 GB, and Linode 4 GB. The 'Linode 2 GB' row is highlighted with a red box. The table columns include Plan, Monthly price, Hourly price, RAM, CPUs, Storage, Transfer, and Network In / Out.

Plan	Monthly	Hourly	RAM	CPUs	Storage	Transfer	Network In / Out
Nanode 1 GB	\$5	\$0.0075	1 GB	1	25 GB	1 TB	40 Gbps / 1 Gbps
Linode 2 GB	\$12	\$0.018	2 GB	1	50 GB	2 TB	40 Gbps / 2 Gbps
Linode 4 GB	\$24	\$0.036	4 GB	2	80 GB	4 TB	40 Gbps / 4 Gbps

© Canseco 2024

# Getting VPL-Jail-System

## 1. Enter your server

- install Visual Studio Code and use the terminal there Open VSCode and press Ctrl-` (i.e. control backtick)
- `ssh root@<your-server-ip>`

## 2. Download the system:

```
wget https://vpl.dis.ulpgc.es/releases/vpl-jail-  
system-4.0.2.tar.gz
```

## 3. Unzip the installer:

```
tar xvf vpl-jail-system-4.0.2.tar.gz
```

# Installing the VPL-Jail-System for the first time:

1. Go to the installer directory

```
cd vpl-jail-system-4.0.2
```

2. Run the installer as root

```
sudo ./install-vpl-sh full start
```

04

# VPL config of Philippine Science High School - MC

# Tips for installing for the first time:

- say yes to **LetsEncrypt**
- put in your **email** when asked for
- type in the given **fully qualified domain name** when asked for (e.g. from Linode)
- say **no** to Kotlin
- say **no** to JGrasp
- then make sure to **accept the connections to VPL** when the browser directs you to

also go to the Local Execution Server, type a space in the box and then hit save.

# If not using cron, run the following every 1 to 2 months

```
systemctl status vpl-jail-system.service  
systemctl restart vpl-jail-system.service
```

```
cd ~  
cd vpl-jail-system-4.0.2  
../install-vpl-sh full noninteractive start
```

- This will renew your Let's Encrypt SSL certificate
- You can run the code sooner if notice any problems with the VPL Jail Server
- Prepare to wait a bit, but the terminal should show you how things are going.

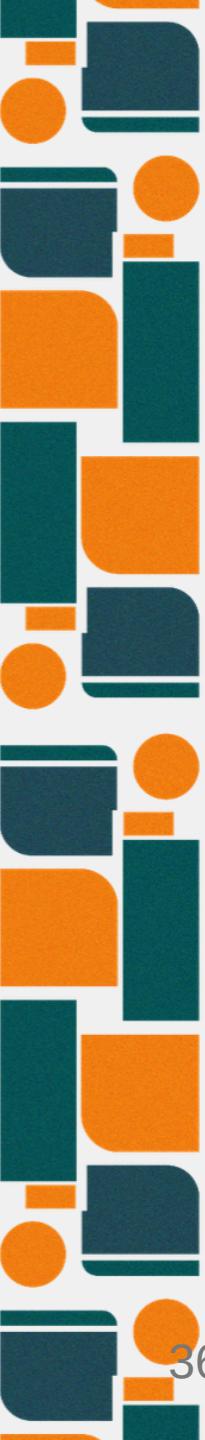
# What it means:

-

```
./install-vpl-sh full noninteractive  
start
```

-

- **full** means to install all the languages it fully supports
- **noninteractive** means to NOT ask you to do anything
- **start** means to start the server once the installation is finished



# Automating daily VPL service restart

- To set up a cron job to restart a service every day on Ubuntu 20.04
  - `crontab -e`
- In the crontab file, to restart the service every day at 3:00 AM, you can use:
- `0 3 * * * systemctl restart vpl-jail-system.service`
- When setting up cron jobs, it's important to ensure that the user has the **right permissions** (e.g. root) to restart the service.

# Automatically restart the server weekly

- The VPL jail server service will automatically run when after the server restarts
- To automatically restart the server every week in Ubuntu 20.04, you can set up a **cron** job to schedule the restart.

```
crontab -e
```

- In the crontab file, add the following line to schedule the server restart every week:

```
0 0 * * 0 reboot
```

# Automatically reinstall the VPL jail system monthly

- To automatically reinstall the VPL jail system from the installer every month in Ubuntu 20.04, you can set up a **cron** job to schedule the reinstall.

```
crontab -e
```

- In the crontab file, add the following line to schedule the system reinstall on the first day of every month:

```
0 0 1 * * cd ~; cd vpl-jail-system-4.0.2; ./install-vpl-sh full noninteractive start
```

no need to do any manual commands at this point

# Add the server address to your list of VPL Execution servers

- We simply use the IP address

```
http://xxx-xxx-xxx-101.ip.linodeusercontent.com  
http://xxx.xxx.xxx.101
```

---

## Execution servers list

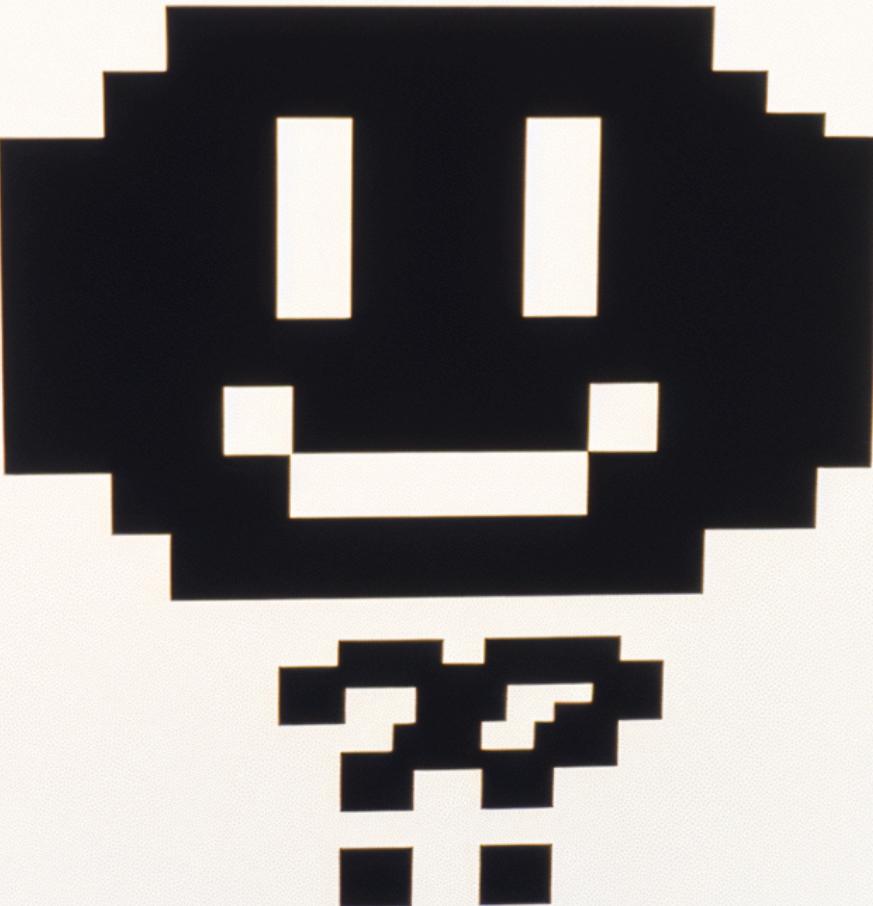
[mod\\_vpl](#) | [jail\\_servers](#)

```
# This server is only for test use.  
# Install your own Jail server and remove the following line  
# http://demojail.dis.ulpgc.es  
http://[REDACTED]-101.ip.linodeusercontent.com  
http://[REDACTED].101
```

VPL should now be  
ready for class. :)



```
ist1: Smidet::[-]_
nde1: bmestect::l>[:/.]...
ist1: Smidet::[-]_
ist1: bmestect::(-)
```



# Thanks!

Do you have any  
questions?

---

[roy@pshs.edu.ph](mailto:roy@pshs.edu.ph)

# Resources

- AI-gen picture: **Leonardo AI**. 2024. <https://app.leonardo.ai/ai-generations>
- AI-gen picture: **Cici AI**. 2024. <https://www.ciciai.com/chat>
- icons: **Font Awesome**. 2024. <https://fontawesome.com/v6/>
- Slides template was created by **Slidesgo** , ported to **Marp**

0X

# Extra slides for Workshop

# Creating Programming Assignments with Moodle VPL

- Step-by-step guide on how to create programming assignments using the plugin
- Defining programming tasks, requirements, and constraints
- Setting up programming languages, libraries, and compiler options
- Configuring evaluation criteria and grading methods
- Demonstrating the use of test cases and expected outputs

# Student Experience with Moodle VPL

- Walkthrough of the student view and interface in the Moodle VPL activity
- Accessing and understanding the programming assignment
- Writing and submitting code within the web-based programming environment
- Running and testing code in the secure sandbox environment
- Receiving immediate feedback and evaluation results

# Best Practices and Tips for Using Moodle VPL

- Sharing effective strategies for designing and implementing programming assignments
- Providing tips for instructors to enhance student engagement and learning outcomes
- Addressing common challenges and how to overcome them
- Highlighting success stories and case studies from other institutions

# Conclusion

- Recap of the key points covered in the talk
- Encouraging further exploration and adoption of the Moodle VPL plugin
- Thanking the audience for their participation and attention