



# Comparison of Students' Programming Skills Before and After Using AI Gemini in Colab with Python

Xayxavath SOUKCHALEUN<sup>1\*</sup>, Vinath MEKTHANAVANH<sup>2</sup>, Thongphet KHONGKETH<sup>3</sup>,  
Bounthanome VILASACK<sup>4</sup>, Sangvang KITTIKHOUN<sup>5</sup>

Office of Administration Division, Souphanouvong University, Lao PDR

**\*Correspondence:** Xaysavath SOUKCHALEUN, Office of Administration Division, Souphanouvong University, Tel +8562052221114, E-mail: soukchaleun.x@su.edu.la

## Abstract

This study investigated the impact of AI Gemini integration in Google Colab on Python programming skills development among 70 computer engineering and information technology students at Souphanouvong University. Using a one-group pretest-posttest design, the research employed programming assessments, time tracking, and observational methods to evaluate skill enhancement. Results demonstrated significant improvements: average programming scores increased from 64.59 to 79.53 (23.1% improvement,  $p \leq 0.001$ ) and average problem-solving time decreased from 47.42 to 30.51 minutes (35.7% reduction). Qualitative analysis revealed enhanced conceptual understanding of programming fundamentals, improved debugging capabilities, and increased student confidence. AI Gemini served as an effective real-time learning assistant, providing immediate feedback and facilitating systematic problem-solving approaches. The findings support AI Gemini integration in programming education as a valuable tool for enhancing learning outcomes, promoting computational thinking skills, and improving programming efficiency. This research contributes to understanding AI-assisted learning environments in computer science education and provides practical implications for programming pedagogy in higher education institutions.

**Keywords:** AI Gemini, Python Programming, AI-assisted Learning, Computational Thinking

## Article Info:

Submitted: November 20, 2025

Revised: December 10, 2025

Accepted: December 18, 2025

## 1. ພາກສະເໜີ

ໃນສະພາບແວດລ້ອມການສຶກສາສາຍກາດິຈິຕອນທີ່ກຳລັງພັດທະນາຢ່າງໄວວາ, ທັກສະການຂຽນໂປແກມຖືກຍອມຮັບວ່າເປັນຄວາມສາມາດທີ່ຈຳເປັນຢ່າງຍິ່ງສຳລັບນັກສຶກສາດ້ານວິສະວະກຳຄອມພິວເຕີ ແລະ ເທັກໂນໂລຊີຂໍ້ມູນຂ່າວສານ. ຢ່າງໃດກໍດີ, ການສອນທັກສະການຂຽນໂປຣແກຣມຍັງໄດ້ປະເຊີນກັບຄວາມທະວີຄວາມຫຍຸ້ງຍາກຫຼາຍ

ດ້ານ, ໂດຍສະເພາະໃນການພັດທະນາຄວາມເຂົ້າໃຈແນວຄິດພື້ນຖານ ແລະ ການນຳໃຊ້ຄວາມຮູ້ໄປແກ້ໄຂບັນຫາຈິງໃນການຂຽນໂປຣແກຣມ (Smith & Johnson (2020).

ຄວາມກ້າວໜ້າທາງດ້ານເທັກໂນໂລຊີປັນຍາປະດິດ (AI) ໃນຊຸມປີຜ່ານໆນີ້, ໂດຍສະເພາະການເກີດຂອງໂຕແບບພາສາຂະໜາດໃຫຍ່ (Large Language Models - LLMs) ເຊັ່ນ Google Gemini.

Liu et al. (2023) ໄດ້ເປີດໂອກາດໃໝ່ໆໃນການແກ້ໄຂບັນຫາດັ່ງກ່າວ. AI Gemini ເປັນເທັກໂນໂລຊີ AI ທີ່ພັດທະນາໂດຍ Google DeepMind ເຊິ່ງເປີດຕົວຢ່າງເປັນທາງການໃນປີ 2023. ໂມເດວນີ້ມີຄວາມສາມາດໃນການປະມວນຜົນຂໍ້ມູນຫຼາກຫຼາຍຮູບແບບທັງຂໍ້ຄວາມ, ຮູບພາບ, ແລະ ເອກະສານ, ພ້ອມທັງສາມາດເຂົ້າໃຈບັນຫາທີ່ສັບສົນ ແລະ ໃຫ້ຄໍາແນະນຳທີ່ຊັດເຈນ ( Demis Hassabis, 2023).

ໃນຂົງເຂດການສຶກສາ, Gemini ຖືກນຳໃຊ້ເປັນເຄື່ອງມືຊ່ວຍໃນການຮຽນຮູ້ທັກສະການຂຽນໂປຣແກຣມ ໂດຍສາມາດໃຫ້ຄໍາແນະນຳທັນທີ, ຊ່ວຍແກ້ໄຂຂໍ້ຜິດພາດ, ແລະ ອະທິບາຍແນວຄິດທາງການຂຽນໂປຣແກຣມໄດ້ຢ່າງຊັດເຈນ ຄຸນສົມບັດພິເສດຂອງ Gemini ປະກອບມີ ການສື່ສານແບບສົນທະນາ ທີ່ເຮັດໃຫ້ການຮຽນຮູ້ເປັນການສົນທະນາທີ່ໃກ້ຊິດ, ການໃຫ້ຄໍາແນະນຳສະເພາະບຸກຄົນ ຕາມຄວາມຕ້ອງການຂອງນັກສຶກສາ, ແລະ ການຊ່ວຍເຫຼືອແບບ Real-time ໃນການແກ້ບັນຫາ (Mekthanavanh et al., 2024).

ການສຶກສາກ່ອນໜ້ານີ້ໄດ້ຊີ້ໃຫ້ເຫັນແນວທາງບວກຂອງການນຳໃຊ້ AI ໃນການສຶກສາ. Zhou & Li (2022) ພົບວ່ານັກສຶກສາທີ່ໄດ້ຮັບການຊ່ວຍເຫຼືອ ແລະ ແນະນຳຈາກ AI ມີຄວາມກ້າວໜ້າໃນການເຂົ້າໃຈແນວຄິດການຂຽນໂປຣແກຣມ. Williams & Brown (2023) ສະແດງໃຫ້ເຫັນວ່າການໃຫ້ຄໍາແນະນຳແບບທັນທີ (Real time) ຈາກ AI ສາມາດຊ່ວຍຫຼຸດເວລາໃນການແກ້ໄຂບັນຫາ ແລະ ເພີ່ມຄວາມມັ່ນໃຈໃຫ້ແກ້ນັກສຶກສາ. Chen et al. (2023) ໄດ້ເນັ້ນໃສ່ຄວາມສຳຄັນຂອງການອອກແບບການໃຊ້ງານ AI ຢ່າງເໝາະສົມໃນຂະບວນການຮຽນຮູ້, ໂດຍສະເພາະໃນການສົ່ງເສີມທັກສະການແກ້ໄຂບັນຫາແບບຍືນຍົງ.

ຢ່າງໃດກໍດີ, ຍັງມີຊ່ອງຫວ່າງທາງດ້ານການຄົ້ນຄວ້າທີ່ກ່ຽວຂ້ອງກັບການນຳໃຊ້ AI ແບບສົນທະນາ, ໂດຍສະເພາະແມ່ນ Google Gemini, ໃນການພັດທະນາທັກສະການຂຽນໂປຣແກຣມຢ່າງເປັນລະບົບ. ການສຶກສາສ່ວນໃຫຍ່ມັກຈະສຸມໃສ່ AI ໃນລັກສະນະຂອງຜູ້ສອນ ຫຼື ເຄື່ອງມືຊ່ວຍສອນ, ແທນທີ່ຈະເປັນ "ຄູ່ຮ່ວມງານ" ໃນຂະບວນການຮຽນຮູ້ຂອງນັກສຶກສາ. Johnson & Lee (2022) ໄດ້ເນັ້ນໃສ່ຄວາມຈຳເປັນໃນການສຶກສາຜົນກະທົບຂອງ AI ທີ່ມີຕໍ່ການພັດທະນາທັກສະດ້ານຕ່າງໆ ຢ່າງເປັນລະບົບ, ໂດຍສະເພາະໃນຂົງເຂດການຂຽນໂປຣແກຣມທີ່ຕ້ອງການທັກສະທີ່ສັບສົນ ແລະ ຫຼາກຫຼາຍ.

ສະນັ້ນ, ການຄົ້ນຄວ້າຄັ້ງນີ້ຈຶ່ງມີຈຸດປະສົງເພື່ອປຽບທຽບຄວາມສາມາດທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາກ່ອນ ແລະ ຫຼັງຈາກໃຊ້ AI Gemini ເປັນເຄື່ອງມືຮ່ວມໃນການຮຽນຮູ້ Python ຢ່າງເປັນລະບົບ. ການຄົ້ນຄວ້າຄັ້ງນີ້ບໍ່ພຽງແຕ່ຈະຊ່ວຍໃຫ້ເຂົ້າໃຈປະສິດທິພາບຂອງ Gemini ໃນການສະໜັບສະໜູນການຮຽນຮູ້, ແຕ່ຍັງຈະນຳໄປສູ່ການພັດທະນາຮູບແບບການສຶກສາທີ່ທັນສະໄໝ ແລະ ມີປະສິດທິພາບສູງຂຶ້ນ. Garcia & Thompson (2023).

## 2. ອຸປະກອນ ແລະ ວິທີການ

ໃນບົດວິໄຈຄັ້ງນີ້ຜູ້ຄົນຄວ້າຈະໄດ້ທົດສອບທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາໂດຍການໃຊ້ກ່ອນ ແລະ ຫຼັງ Gemini AI ເພື່ອທົດສອບວັດລະດັບຄວາມຮູ້ຄວາມສາມາດໃນທັກສະການຂຽນ

ໂປຣແກຣມ ເຊິ່ງພວກເຮົາໄດ້ຂຽນໂຄດທີ່ກ່ຽວ ກັບການປະຕິບັດການທົດລອງເຊັ່ນ: ການສ້າງໂຄ້ດ, ການສ້າງ pseudocode ຂອງອັງກິດສິລາຈາກ text, ການກວດແກ້ໂຄ້ດພ້ອມທາງອະທິບາຍໂຄ້ດ ແລະ ແນວທາງການເຜີຍນຳຈາກ AI Gemini . ນອກຈາກນັ້ນ ຍັງໄດ້ສຳຫຼວດ ແລະ ປະຕິບັດການທົດລອງ Gemini ກັບນັກສຶກສາ ແລະ ອາຈານສອນ ເພື່ອສະແດງໃຫ້ເຫັນວິທີ ການນຳໃຊ້ເຄື່ອງມືນີ້ ເພື່ອສະໜັບສະໜູນທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາໂດຍມີເຄື່ອງມື AI ຄື Gemini.

### 2.1 ປະຊາກອນ ແລະ ກຸ່ມຕົວຢ່າງ

ກຸ່ມຕົວຢ່າງປະກອບດ້ວຍນັກສຶກສາປີທີ 2 ສາຂາວິສະວະກຳຄອມພິວເຕີ ແລະ ສາຂາວິສະວະກຳອຸທິ ພາກວິຊາວິສະວະກຳຄອມພິວເຕີ ຄະນະວິສະວະກຳສາດ ມະຫາວິທະຍາໄລ ສຸພານຸວົງ ຈຳນວນ 70 ຄົນ ເຊິ່ງເປັນຕົວແທນທີ່ຖືກຄັດເລືອກມາດ້ວຍວິທີການສຸເອົາແບບສະເພາະ (Purposive Sampling) Etikan et al. (2016). ແກນຫຼັກໃນການຄັດເລືອກປະກອບມີ: (1) ຜ່ານການຮຽນວິຊາ Python Programming ມາແລ້ວ, (2) ມີຄະແນນພື້ນຖານວິຊາຄອມພິວເຕີຢູ່ໃນລະດັບກາງໆໃນຫ້ອງຮຽນດ່ຽວກັນ.

### 2.2 ເຄື່ອງມືທີ່ໃຊ້ໃນການຄົ້ນຄວ້າ

ເຄື່ອງມືຫຼັກທີ່ນຳໃຊ້ໃນການຄົ້ນຄວ້າແມ່ນ ການທົດສອບທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາ ເຊິ່ງຖືກສ້າງຂຶ້ນເພື່ອວັດລະດັບຄວາມສາມາດດ້ານທັກສະການຂຽນໂປຣແກຣມພາສາ Python ໃນ Google Colab ຂອງນັກສຶກສາ. ການທົດສອບນີ້ປະກອບດ້ວຍບັນຫາ, ວິທີການແກ້ໄຂຂໍ້ຜິດພາດ ແລະ ແນວທາງການຂຽນໂປຣແກຣມຢ່າງມີໄລຍະທີ່ສຳຄັນເຊັ່ນ: ການໃຊ້ງານຝັງຊັ້ນ, ການຄວບຄຸມການເຮັດວຽກດ້ວຍລຸບ, ການສ້າງເງື່ອນໄຂ ແລະ ການຈັດການຂໍ້ມູນ. ໂດຍມີການທົດສອບກ່ອນ ແລະ ຫຼັງ ໃຊ້ AI ຄື Gemini ເຂົ້າໃນການສອນ.

### 2.3 ວິທີການເກັບກຳຂໍ້ມູນ

ການເກັບກຳຂໍ້ມູນແມ່ນໄດ້ມີການທົດສອບນັກສຶກສາກຸ່ມທີ່ມີພື້ນຖານທັກສະການຂຽນໂປຣແກຣມໃນລະດັບຄ້າຍໆກັນໄດ້ມີການທົດສອບນັກສຶກສາກ່ອນ ແລະ ຫຼັງໃຊ້ AI ຄື Gemini ຮອບດ້ານ ແລະ ເປັນລະບົບ ໄດ້ປະກອບມີ ການບັນທຶກຄະແນນການປະເມີນກ່ອນ ແລະ ຫຼັງການໃຊ້ເຄື່ອງມື ການຈັບເວລາໃນການແກ້ໄຂບັນຫາໂດຍໃຊ້ເຄື່ອງມືຈັບເວລາແບບອັດຕອໂນມັດ, ການບັນທຶກຂໍ້ມູນການສັງເກດການກ່ຽວກັບຂະບວນການດິບກ ແລະ ການແກ້ໄຂບັນຫາ, ແລະ ການວິເຄາະຮູບແບບການສື່ສານຂອງນັກສຶກສາກັບ AI Gemini ເຊິ່ງໃຊ້ເວລາສອນ 32 ຊົ່ວໂມງອາທິດລະ 4 ຊົ່ວໂມງ ໃນການທົດສອບທັງໆໝົດ.

### 2.4 ການວິເຄາະຂໍ້ມູນ

ການວິເຄາະຂໍ້ມູນແມ່ນນຳເອົາຄະແນນ ແລະ ການຈັບເວລາໃນການປະຕິບັດຕົວຈິງຂອງນັກສຶກສາເພື່ອນຳວິເຄາະໂດຍໃຊ້ ໂປຣແກຣມ Python ທີ່ສ້າງຂຶ້ນເອງ. ຂະບວນການວິເຄາະຂໍ້ມູນປະກອບມີ ການ

ຄິດໄລ່ຈຳນວນກສິກສາພ້ອມອະທິບາຍເພື່ອຫາຄ່າ (ຄ່າສະເລ່ຍ, ຄ່າສ່ວນກ່ອນມາດຕະຖານ), ການທົດສອບ t ຄູ່ (Paired t-test) ສຳລັບການປຽບທຽບຜົນກ່ອນ ແລະ ຫຼັງການທົດລອງການວິເຄາະຄວາມແຕກຕ່າງລະຫວ່າງດ້ວຍການວິເຄາະຄວາມແຕກຕ່າງ (ANOVA). ວິທີການວິເຄາະເຫຼົ່ານີ້ຊ່ວຍໃຫ້ສາມາດຕີຄວາມຫມາຍຂໍ້ມູນໄດ້ຢ່າງລະອຽດ ແລະ ມີເຫດຜົນ

### 3. ຜົນໄດ້ຮັບ

ຜານການທົດລອງນຳໃຊ້ AI Gemini ເຂົ້າໃນການຂຽນໂປແກມ ພົບວ່າ ຄຸນນະພາບການຂຽນໂປແກມແມ່ນມີການເພີ່ມຂຶ້ນຢ່າງຊັດເຈນຈາກການປຽບທຽບຜົນການທົດສອບກ່ອນ ແລະ ຫຼັງການນຳໃຊ້ AI Gemini ຄື ຄະແນນສະເລ່ຍຜົນທົດສອບກ່ອນເທົ່າກັບ 64.59 ແລະ ຄະແນນສະເລ່ຍຜົນທົດສອບຫຼັງ ເທົ່າກັບ 79.53 ເຫັນວ່າມີການປັບປຸງດີຂຶ້ນ 23.1% ແລະ ມີຄວາມສຳພັນທາງສະຫຼັບ (p < 0.001) (ຕາຕະລາງ 1). ນອກຈາກນີ້, ປະສິດທິພາບໃນການແກ້ໄຂບັນຫາກໍໄດ້ຮັບການພັດທະນາຫຼາຍກວ່າເກົ່າ ໂດຍໄດ້ໃຊ້ເວລາໃນການແກ້ໄຂບັນຫາຫຼຸດລົງເຖິງ 35.7% ຈາກ 47.42 ນາທີ ເຫຼືອພຽງ 30.51 ນາທີ. ການຫຼຸດລົງຂອງຄ່າ ມາດຖານ ໃນທຸກຕົວຊີ້ວັດຍັງສະແດງໃຫ້ເຫັນຄວາມປະຫຍັດເວລາໃນການຂຽນໂປແກມໄດ້ດີຫຼາຍ (ຕາຕະລາງ 2).

ທັກສະການແກ້ດິບັກ ແລະ ການແກ້ໄຂຂໍ້ຜິດພາດໄດ້ຮັບການປັບປຸງຢ່າງຫຼວງຫຼາຍ ເຊິ່ງເຫັນໄດ້ຈາກການຫຼຸດລົງຂອງຈຳນວນຂໍ້ຜິດພາດສະເລ່ຍເຖິງ 60.3% ແລະ ການເພີ່ມຂຶ້ນຂອງຄວາມສາມາດໃນການແກ້ໄຂບັນຫາເຖິງ 58.6%. ຜົນການປັບປຸງເຫຼົ່ານີ້ບໍ່ພຽງແຕ່ສະແດງເຖິງການພັດທະນາທັກສະດ້ານວິຊາການເທົ່ານັ້ນ ແຕ່ຍັງສະທ້ອນເຖິງການພັດທະນາຄວາມໝັ້ນໃຈໃນທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາອີກດ້ວຍ.

### 4. ການວິພາກຜົນ

ຜົນການວິເຄາະຂໍ້ມູນຈາກການທົດສອບກ່ອນ ແລະ ຫຼັງການນຳໃຊ້ AI Gemini ເປັນເຄື່ອງມືຊ່ວຍໃນການຮຽນຮູ້ໄດ້ສະແດງໃຫ້ເຫັນການປ່ຽນແປງທີ່ມີເຄື່ອງ AI Gemini ມາຊ່ວຍທາງດ້ານທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາ. ຜົນການທົດສອບຄັ້ງນີ້ສະແດງເຖິງການເພີ່ມຂຶ້ນຂອງຄະແນນໄດ້ຢ່າງຊັດເຈນໂດຍຄະແນນສະເລ່ຍ (M = 79.53, SD = 6.15) ທຽບກັບກ່ອນການນຳໃຊ້ (M = 64.59, SD = 8.24), ຄິດເປັນອັດຕາການພັດທະນາ 23.1%. ຍັງຊີ້ບອກເຖິງຄວາມເປັນເອກະພາບທີ່ສູງຂຶ້ນໃນການເຂົ້າໃຈແນວຄິດທັກສະການຂຽນໂປຣແກຣມຂອງນັກສຶກສາໃນກຸ່ມຕົວຢ່າງ ເຊິ່ງສອດຄ່ອງກັບບົດຄົ້ນຄວ້າຂອງ Yilmaz et al. (2023) ພົບວ່າມີການປັບປຸງ 22%.

ດ້ານປະສິດທິພາບໃນການແກ້ໄຂການຫຼຸດຜ່ອນເວລາເຊິ່ງໄດ້ສະເລ່ຍຫຼັງການໃຊ້ AI Gemini ໄດ້ຫຼຸດລົງຢ່າງຫຼວງຫຼາຍຈາກ 47.42 ນາທີ (SD = 12.36) ເປັນ 30.51 ນາທີ (SD = 8.47), ຄິດເປັນການປັບປຸງ 35.7% ຜົນການປ່ຽນແປງນີ້ ເຊິ່ງສອດຄ່ອງກັບບົດຄົ້ນຄວ້າຂອງ Hasan et al. (2025) ພົບການປັບປຸງ 34%.

ດ້ານທັກສະການແກ້ໄຂດິບັກ ແລະ ການແກ້ໄຂຂໍ້ຜິດພາດໄດ້ຮັບການປັບປຸງຢ່າງຫຼວງຫຼາຍ ເຊິ່ງເຫັນໄດ້ຈາກການຫຼຸດລົງຂອງຈຳນວນຂໍ້ຜິດພາດສະເລ່ຍຈາກ 5.8 ຂໍ້ຜິດພາດ ເປັນ 2.3 ຂໍ້ຜິດພາດ, ຄິດເປັນການຫຼຸດລົງ 60.3% ຈາກການສຶກສານີ້ ເຫັນວ່າມີການປັບປຸງສູງກວ່າຜົນການສຶກສາຂອງ Ahmed et al. (2020) ໄດ້ພົບເຫັນການປັບປຸງພຽງແຕ່ 50%. ຄວາມສາມາດໃນການແກ້ໄຂບັນຫາເອງຂອງນັກສຶກສາໄດ້ເພີ່ມຂຶ້ນຈາກ 34.2 ມາເປັນ 92, ຄິດເປັນການພັດທະນາ 58.6%. ຜົນການສຶກສານີ້ມີຄວາມໄກ້ຄຽງກັບຜົນການສຶກສາ ຂອງ Taylor & Martin. (2019) ໄດ້ພົບການປັບປຸງພຽງແຕ່ 54%.

ຜົນການສຶກສາສະແດງໃຫ້ເຫັນວ່າຫຼັງການນຳອີ້ AI Gemini ມາຊ່ວຍໃນການພັດທະນາທັກສະການຂຽນໂປຣແກຣມເຫັນໄດ້ຊັດເຈນ ແລະ ຄະແນນດີຂຶ້ນ 23.1% ເຊິ່ງສອດຄ່ອງກັບ ທີ່ໄດ້ສະຫຼຸບລາຍງານກ່ຽວກັບປະສິດທິພາບຂອງ AI ການສະໜັບສະໜູນເຂົ້າໃນການສຶກສາຮຽນຮູ້ທັກສະການຂຽນໂປຣແກຣມ. ບໍ່ພຽງແຕ່ເທົ່ານີ້, ການສຶກສາຂອງພວກສາມາດເຮັດໃຫ້ປະຫຍັດເວລາ (ຫຼຸດລົງ 35.7%) ເຊິ່ງໄກ້ຄຽງກັບຜົນການສຶກສາຂອງ Hasan et al. (2025). ຜົນການປະຫຍັດເວລາ (ຫຼຸດລົງ) 34%.

ນອກຈາກນີ້ນັກສຶກສາຍັງສາມາດຂຽນໂປຣແກຣມໂດຍສະເພາະແມ່ນ ການແກ້ດິບັກ ແລະ ການແກ້ໄຂຂໍ້ຜິດພາດເຊິ່ງໄດ້ຮັບການຍົກລະດັບຂຶ້ນຢ່າງຊັດເຈນພາຍຫຼັງການໃຊ້ AI Gemini ເປັນເຄື່ອງມືຊ່ວຍຮຽນ, ໂດຍໄດ້ເຫັນຈາກ ຂໍ້ຜິດພາດຫຼຸດລົງ 60.3%. ພ້ອມກັນນີ້ນັກສຶກສາຍັງສາມາດແກ້ໄຂຂໍ້ຜິດພາດໃນການຂຽນໂປຣແກຣມດ້ວຍຕົນເອງຢ່າງວ່ອງໄວເຖິງ 58.6 %

### 5. ສະຫຼຸບ

ການໃຊ້ AI Gemini ໃນ Google Colab ສາມາດຊ່ວຍເພີ່ມປະສິດທິຜົນການຮຽນຮູ້ທັກສະການຂຽນໂປຣແກຣມ ພາສາ Python ຂອງນັກສຶກສາໄດ້ດີຫຼາຍ ແລະ ປະຫຍັດເວລາໃນການແກ້ໄຂບັນຫາ, ພ້ອມກັນນີ້ ການແກ້ດິບັກ ແລະ ການແກ້ໄຂຂໍ້ຜິດພາດ ຫຼຸດລົງຈຳນວນຫຼາຍ. ຈາກຜົນການສຶກສາ ສະແດງໃຫ້ເຫັນການປັບປຸງຢ່າງມີຄວາມສຳຄັນ ແລະ ພັດທະນາທັກສະການຂຽນໂປແກມ, ປະສິດທິພາບໃນການແກ້ໄຂບັນຫາ, ຄວາມເຂົ້າໃຈແນວຄິດ, ແລະ ຄວາມສາມາດໃນການດິບັກ.

AI Gemini ມີບົດບາດເປັນ "ຜູ້ຊ່ວຍສອນອັດສະລິຍະ" ໂດຍການໃຫ້ຄຳແນະນຳແບບທັນທີ ແລະ ການວິເຄາະຂໍ້ຜິດພາດ. ນັກສຶກສາໄດ້ພັດທະນາທັກສະການຄິດຄຳນວນ ແລະ ວິທີການແກ້ໄຂບັນຫາແບບມີລະບົບ. ສະນັ້ນ, ເທັກໂນໂລຊີ AI ໃນການສຶກສາ ມີບົດບາດສິ່ງເສີມການຮຽນຮູ້ທີ່ດີກວ່າ.

ດັ່ງນັ້ນ, ການໜັບສະໜູນການນຳໃຊ້ AI Gemini ໃນການສຶກສາທັກສະການຂຽນໂປຣແກຣມ ໃນຂະນະທີ່ເນັ້ນໃສ່ຄວາມຕ້ອງການຂອງການຄົ້ນຄວ້າເພີ່ມເຕີມກ່ຽວກັບການພັດທະນາທັກສະໃນໄລຍະຢ່າວ ແລະ ໃຫ້ສາມາດນຳໃຊ້ໃນຫຼາຍຂົງເຂດວຽກງານ.

### 6. ຂໍ້ຂັດແຍ່ງ

ຂ້າພະເຈົ້າໃນນາມຜູ້ຄົ້ນຄວ້າວິທະຍາສາດ ຂໍປະຕິຍານຕີນວ່າ ຂໍ້ມູນທັງໝົດທີ່ມີໃນບົດຄວາມວິຊາການດັ່ງກ່າວນີ້ ແມ່ນບໍ່ມີຂໍ້ຂັດແຍ່ງທາງຜົນປະໂຫຍດກັບພາກສ່ວນໃດ ແລະ ບໍ່ໄດ້ເອື້ອປະໂຫຍດໃຫ້ກັບ

ພາກສ່ວນໃດພາກສ່ວນໜຶ່ງ, ກໍລະນີມີການລະເມີດ ໃນຮູບການໃດໜຶ່ງ ຂ້າພະເຈົ້າມີຄວາມຍິນດີ ທີ່ຈະຮັບຜິດຊອບແຕ່ພຽງຜູ້ດຽວ.

## 7. ເອກະສານອ້າງອີງ

- Ahmed, U., Zaidman, S., & van Deursen, A. (2021). AI-powered bug detection and fixing in programming education: A systematic review. *ACM Computing Surveys*, 54(6), 1-36.
- Ahmed, S., Li, Y., & Kumar, V. (2020). AI-assisted debugging tools and their impact on novice programmers. *Journal of Educational Computing Research*, 58(6), 1123–1145.
- Chen, X., Liu, Y., Wang, Z., & Zhang, H. (2023). AI integration in programming education. *Journal of Educational Technology*, 45(2), 123-145.
- Elnaffar, S., Rashidi, F., & Abualkishik, A. Z. (2025). Teaching with AI: A systematic review of chatbots, generative tools, and tutoring systems in programming education
- Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American Journal of Theoretical and Applied Statistics*, 5(1), 1-4.
- Garcia, M., & Thompson, R. (2023). *AI tools in computational thinking development*. MIT Press.
- Google. (2023). *Gemini AI assistant documentation*. Google AI Publications. Retrieved from: <https://blog.google/technology/ai/google-gemini-ai/#sundar-note>.
- Johnson, R., & Lee, S. (2022). AI impacts on skill development. *Computing Education Research*, 15(3), 234-256.
- Huang, W., Hew, K. F., & Fryer, L. K. (2022). Chatbots for language learning: Are they really useful? A systematic review of chatbot-supported language learning. *Journal of Computer Assisted Learning*, 38(1), 237-257.
- Hasan Amiri, S. M., & Islam, M. M. (2025). Enhancing Python programming education with an AI-powered code helper: Design, implementation, and impact
- Liu, W., Chen, H., & Rodriguez, A. (2023). Comparative analysis of large language models in computer science education. *Computers & Education*, 195, 104721.
- Mekthanavanh, V., Meksavanh, B., & Bouddy, S. (2024). ChatGPT Enhances Programming Skills of Computer Engineering Students. *Souphanouvong University Journal of Multidisciplinary Research and Development*, 10(1), 1-9.
- Smith, J., & Johnson, K. (2020). Programming skill assessment in higher education. *Journal of Computer Science Education*, 15(2), 45-62.
- Taylor, S., & Martin, P. (2019). Time management in programming problem-solving. *International Journal of Educational Technology*, 8(3), 112-125.
- Williams, R., & Brown, T. (2023). Real-time AI guidance in programming education. *AI in Education Journal*, 12(1), 67-89.
- Yilmaz, R., & Karaoglan Yilmaz, F. G. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers & Education: Artificial Intelligence*, 4, 100147.
- Zhou, L., & Li, M. (2022). AI assistance in programming concept understanding. *Computer Science Education Review*, 18(2), 156-178.

ຕາຕະລາງ 1. ການປັບປຸງທັກສະການຂຽນໂປຣແກມຂອງນັກສຶກສາ

ຕົວຊີ້ວ່າການປະເມີນ	ກ່ອນທົດສອບ	ຫຼັງທົດສອບ	ການປັບປຸງ	ຄ່າ p
<p>ຄະແນນສະເລ່ຍ (%)</p> 	64.59 ± 8.23	79.53 ± 6.45	+23.1%	<0.001
ເວລາແກ້ໄຂບັນຫາ (ນາທີ)	47.42 ± 12.36	30.51 ± 8.74	-35.7%	<0.001
ຈຳນວນຂໍ້ຜິດພາດສະເລ່ຍ	5.8 ± 2.1	2.3 ± 1.4	-60.3%	<0.001
ຄວາມສາມາດແກ້ໄຂບັນຫາເອງ (%)	45.6 ± 12.8	72.3 ± 9.5	+58.6%	<0.001

## ຮູບທີ່ 1 ການແກ້ໄຂໂຄດ (ດີບັກ) ໂດຍ ການນຳໃຊ້ AI Gemini

```
python Copy Download  
  
# ສ້າງ Object ແລະ ທົດສອບ  
student1 = Student("ສິມພອນ", [75, 80, 85, 90, 78])  
student2 = Student("ວິໄລ", [60, 65, 70, 55, 68])  
  
print(f"ນັກສຶກສາ: {student1.name}")  
print(f"ຄະແນນ: {student1.scores}")  
print(f"ເລກສະເລ່ຍ: {student1.calculate_average():.2f}")  
print(f"ຜົນການຮຽນ: {student1.get_grade()}")  
  
print(f"\nນັກສຶກສາ: {student2.name}")  
print(f"ຄະແນນ: {student2.scores}")  
print(f"ເລກສະເລ່ຍ: {student2.calculate_average():.2f}")  
print(f"ຜົນການຮຽນ: {student2.get_grade()}")
```

## ຮູບທີ່ 2 ການແກ້ໄຂໂຄດ (ດີບັກ) ຮູບແບບລຸບໂດຍການນຳໃຊ້ AI Gemini

```
1 # ການທົດສອບໂຄດທີ່ແກ້ໄຂແລ້ວ  
2 def test_functions():  
3     # ທົດສອບຜົນລວມຂອງລຸ້ນໄວ້  
4     numbers = [1, 2, 3, 4, 5]  
5     result_sum = calculate_sum(numbers)  
6     print(f"ຜົນລວມຂອງ {numbers} = {result_sum}")  
7  
8     # ທົດສອບຜົນລວມກວດສອບຄູ່/ຄືກ  
9     test_numbers = [4, 7, 10, 15]  
10    for num in test_numbers:  
11        result = check_even_odd(num)  
12        print(f"ຕົວເລກ {num} ແມ່ນ: {result}")  
13  
14    # ທົດສອບຜົນລວມຫາຄ່າສູງສຸດ  
15    max_value = find_max(10, 25, 15)  
16    print(f"ຄ່າສູງສຸດຈາກ 10, 25, 15 ແມ່ນ: {max_value}")  
17  
18 # ເຮັນໃຊ້ງານຜັງຊັ້ນ  
19 test_functions()  
  
*** ຜົນລວມຂອງ [1, 2, 3, 4, 5] = 15  
ຕົວເລກ 4 ແມ່ນ: Even  
ຕົວເລກ 7 ແມ່ນ: Odd  
ຕົວເລກ 10 ແມ່ນ: Even  
ຕົວເລກ 15 ແມ່ນ: Odd  
ຄ່າສູງສຸດຈາກ 10, 25, 15 ແມ່ນ: 25
```

### ຮູບທີ່ 3 ການອະທິບາຍໂຄດ (ດີບັກ) ໂດຍການນຳໃຊ້ AI Gemini

```
1 # ຄຳຖາມຂອງນັກສຶກສາໃຫ້ AI:
2 """
3 ຊ່ວຍກວດແກ້ໂຄດ Python ນີ້ໃຫ້ຖືກຕ້ອງ:
4 1. ພັງຊັນຄິດໄລ່ຜົນລວມບໍ່ຖືກຕ້ອງ
5 2. ພັງຊັນກວດສອບຄູ່/ຄືກມີ error
6 3. ພັງຊັນຫາຄ່າສູງສຸດຂຽນບໍ່ສົມບູນ
7 """
8
9 # ຄຳຕອບຈາກ AI Gemini (ໂຄດທີ່ແກ້ໄຂແລ້ວ):
10 def calculate_sum(numbers):
11     """
12     ຄິດໄລ່ຜົນລວມຂອງຕົວເລກໃນ list
13     """
14     total = 0
15     for num in numbers: # ແກ້: ໃຊ້ for num in numbers ແທນ for i in range
16         total += num
17     return total
18
19 def check_even_odd(num):
20     """
21     ກວດສອບຕົວເລກຄູ່ ຫຼື ຄືກ
22     """
23     if num % 2 == 0: # ແກ້: ໃຊ້ == ແທນ =
24         return "Even"
25     else:
26         return "Odd" # ແກ້: ເພີ່ມ :
27
28 def find_max(a, b, c):
29     """
```

### ຮູບທີ່ 4 ການສືບທຽບການຂຽນໂຄດ (ດີບັກ) ຂອງນັກສຶກ ແລະ AI Gemini

```
# ຂໍ້ຜິດພາດທີ່ພົບເຫັນ:
"""
1. ການຂາດພາກວາມີຕົວ 'self' ໃນ methods
2. ການໃຊ້ operator '=' ແທນ '==' ໃນການບຽບທຽບ
3. ການຂາດ colon (:) ຫຼັງຈາກ condition
4. ການຂາດຄຳສັ່ງ return ໃນ function
5. ການໃຊ້ຕົວແປທີ່ບໍ່ຖືກຕ້ອງ
6. ການບໍ່ກວດສອບຂໍ້ມູນກ່ອນນຳໃຊ້
"""

# ຄຳແນະນຳຈາກ AI ສຳລັບນັກສຶກສາ:
"""
1. ກ່ອນຂຽນໂຄດ ໃຫ້ວາງແຜນກ່ອນ
2. ຂຽນໂຄດຄ່ອຍໆ ແລະ ທົດສອບໄປດ້ວຍ
3. ອ່ານ error message ໃຫ້ອອກ
4. ໃຊ້ print() ເມື່ອກວດສອບຄ່າຕົວແປ
5. ຂໍຄວາມຊ່ວຍເຫຼືອຈາກ AI ເມື່ອຕິດບັນຫາ
"""
```

### ຮູບທີ່ 5 ຂໍ້ຜິດພາດທີ່ພົບເຫັນ ແລະ ຄຳແນະນຳຈາກ AI Gemini