

千人狂歡 最受歡迎的程式課

神祕課號 CS61A、CS10，彷彿一串密碼，從此引爆了柏克萊學生的電腦科學狂潮。是什麼樣的魅力，讓上千名「程式菜鳥」排隊上課？

下午兩點不到，加州大學柏克萊分校的薩特大門旁，人潮已經佔據學生活動中心內的所有座位，大廳也站滿了排隊人潮，有學生索性坐在活動中心外的階梯上——他們都在等待。

這門讓學生心甘情願提早到、甚至排隊等候的課，叫作「電腦程式結構與釋義」，是全柏克萊最受歡迎的課，俗稱「CS61A」。去年光是修課人數有兩千五百人；今年三千人，等同一半的柏克萊學生都上這門課。但五年前，這門課只有五百個學生。

掀起柏克萊學程式熱潮的，還有另一堂課號 CS10 的「程式的美麗與樂趣」，是程式設計入門課程，改良麻省理工學院的程式教學軟體 Scratch，用圖像程式語言 Snap! 教學。

今年吸引三百人修課，甚至顛覆傳統性別刻板印象，超過一半的修課學生都是女生。由於淺顯易懂，更被美國列入高中先修課程的教材。

Party with Thousands: The most popular programming courses

At Berkeley, students' obsession with computer science has been ignited by a duo of courses with mysterious and cryptic course codes: CS61A and CS10. What compels thousands of programming rookies to line up for class?

It's just before 2PM at UC Berkeley, and around Sather Gate, crowds have already filled every seat inside the Student Union. The lobby, too, is filled with people standing in line, and some students simply sit on the steps outside—they are all waiting.

The course is called The Structure and Interpretation of Computer Programs, and it's one for which students are willing to arrive early and line up for. It's the most popular course at UC Berkeley, commonly referred to as CS61A. Last year, 2500 students took the course; this year it's 3000, or about half of the Berkeley student population. Just five years ago, there were only 500 students in the course.

The other course responsible for the CS mania at Berkeley is CS10: The Beauty and Joy of Computing. It's an introductory course with a visual programming language called Snap!, an improved version of the Scratch language developed at MIT.

This year there are 300 students enrolled, and defying the stereotypical gender ratios, over half of the students are female. The course, designed to be easy to understand, has even been adopted into the Advanced Placement curriculum of American high schools.

向非本科生遞出橄欖枝

這兩門課程在柏克萊都屬於低階入門課程。根據柏克萊校園課程報告，修課人數增加主要是來自非主修電腦科學的學生。在矽谷，電腦識讀逐漸被視為大學生核心學術與工作能力，以因應求才若渴的科技公司與新創產業。

CS61A 老師、加州大學柏克萊分校電機工程與電腦科學系 (EECS) 教授德內洛 (John DeNero)，過去在 Google 負責 Google 翻譯的開發，拋下高薪開課，就是認為每個學生都應具備基礎的程式能力，跟電腦一樣思考，培養解決問題的能力。

「理解現在的世界需要不同工具，我可以改變學生的未來跟職業選擇，」他強化語氣，希望這門課有賦權的功能。

「讓學生先對程式產生興趣，再發現可以用程式做出真實的東西，」CS10 老師、加州大學柏克萊分校電機工程與電腦科學系教授葛西亞 (Dan Garcia) 向前傾，忙著示範如何用 Snap! 畫出持續盛放、不停旋轉的多邊形花朵。

這兩門課吸引大量非主修電腦科學的學生，同樣追求從做中學的理念，但教學方法截然不同。

Extending an Olive Branch to Non-CS Students

Both courses are lower-division introductory classes at Berkeley. According to Cal's course reports, most of increased enrollment comes from non-CS majors. In Silicon Valley, where tech companies and startups are hungry for talent, computer literacy is becoming a vital academic and technical skill.

CS61A is taught by Berkeley Electrical Engineering and Computer Science (EECS) Professor John DeNero, formerly responsible for the research and development of Google Translate. He left his high-salaried position for academia, believing that every student should be equipped with the basic programming skills and computational thinking to logically solve problems.

“To understand today’s world, we need a different set of tools. I can change my students’ future and their career choices,” he says, emphasizing his hopes that the course will be empowering.

CS10 is taught by Berkeley EECS Professor Dan Garcia. “Allow the students to become interested in programming, then they’ll discover that they can build real things with programs,” he says, leaning forward as he demonstrates how Snap! can be used to create a multi-petal flower that continuously blooms and spins.

Both courses attract large amounts of non-CS majors with the same philosophy of hands-on-learning, but the pedagogy is wildly different.

專案實作 立即給予成就感

在 CS61A，三千個學生同時上課，令人難以想像。開學第一堂課，甚至塞爆全校最大的禮堂，於是德內洛開放學生在家裡遠端上課。

對他而言，學程式重點不在授課，而是每堂課後、每週至少三小時起跳的實驗和討論。

CS61A 教三種程式語言，包括現在最常用的 Python。德內洛認為，學程式語言需要非常多實作練習。又因為絕大部份的學生都是非電腦科學相關科系，要讓學生不受到打擊，最好的做法，就是讓學生可以看到立即成果，同時建構可以支持學生繼續學習的環境。

因此，這門課有兩百五十位助教，輪流帶每週五十個分組討論和實驗時段，甚至也有線上回答跟排程的機制，也追蹤回覆問題的時間，確保每個學生的問題能隨時被解答。

走進柏克萊蘇打教學樓三樓的實驗室，三十個學生對著電腦敲打鍵盤，有的交頭接耳討論遊戲設計的介面。今年畢業、即將到 Google 擔任工程師的助教葛林諾就坐在實驗室最前面，邊忙著安排週末的複習時段，邊等著學生舉手發問。

Projects Provide an Immediate Sense of Accomplishment

It's hard to imagine taking CS61A with 3000 other students. Since the first lecture overflowed the school's biggest auditorium, DeNero has begun allowing students to watch videos at home.

For him, the key to learning programming isn't lectures, but rather the weekly 3+ hours of lab and discussion that occur after each lesson.

CS61A teaches three different programming languages, including one of today's most widely used ones—Python. DeNero believes that in order to learn a programming language, one needs lots of hands-on practice. Because most students are from majors unrelated to CS, the best way for students to avoid frustration and to build a supportive learning environment is to have them see the immediate results of their work.

Hence, this course has 250 teaching assistants, each leading one of the 50 weekly discussion and lab sections. There is even a schedule for TA's to answer questions online and track the response time, ensuring that every student's question is answered in a timely manner.

In the third floor labs of Berkeley's Soda Hall, around 30 students are tapping away on the keyboard, and some of them are discussing the interface for a game design. Tim Greeno, a graduating TA who is headed to Google as a software engineer is sitting at the front of the lab, busily organizing weekend review sessions and waiting to take student questions.

和傳統電腦科學教育不同，CS61A 主打專案式教學。德內洛認為，維持學習的動機就是不讓學生失望，讓學生看到可能性。

除了每週作業，兩到三週就有個專案實作，期末則會有一個大型專題，「讓學生發現，原來他們可以用這些工具，去完成過去從沒想過可以完成的事情，」德內洛說。

舉例來說，開學第四週學生就要做出美食地圖推薦 app，自動推薦、搜尋使用者曾評價過的類似餐廳；第六週學生就已經分組寫出網頁遊戲。

今年大一的瓦夏也坦言，作業負擔很重，卻也認為透過實作訓練，更習慣將大問題解構成小問題思考的流程。

每到學期末，德內洛總習慣問一個問題：「是否對電腦科學更有興趣？」超過八五%的學生，願意將電腦科學列為主修選項之一。

儘管每個修過 CS61A 這門課的學生都說難度高，但德內洛從沒想過讓這門課變簡單，「開一門令人挫折的課，比開心的課還容易，但要從挫折中看到進步，運用電腦運算思惟解決全新的問題。」

In contrast to a traditional computer science education, CS61A is project-based. According to DeNero, the best way to continuously motivate students is to not let them experience disappointment, and instead see the possibilities for CS.

Aside from weekly homework assignments, projects are assigned every two to three weeks. Near the end of the semester, there is a capstone project. DeNero says, “The students will discover that they can actually use these tools to accomplish things they never thought were possible.”

For example, students were asked in the fourth week of class to create a map-based restaurant recommendation app that recommends and searches for similar restaurants based on a users' prior reviews. By the sixth week, student teams were creating a web-based game.

Freshman Washa says that although the academic load is heavy, the project training has made her/him more comfortable with breaking down big problems into a series of smaller tasks.

At the end of every semester, DeNero always asks students the same, “Are you more interested in computer science?” Over 85% of students are willing to consider computer science as a possible major.

Even though every student who's ever taken CS61A says the course is difficult, DeNero has never considered making the curriculum easier. “Teaching a challenging course is easier than a happy one, but these challenges bring progress, and teaches students to use computational logic to solve brand new problems.”

抓住創意的一瞬

相較於學生眼中太過困難的 CS61A，CS10 更是針對非電腦科系主修的學生，主打透過好玩的學習讓學生有興趣，觸發學習程式的契機。

這堂課教的程式開發軟體 Snap!，不同於過去一般認知一行又一行的文字程式碼，而是將程式碼圖像化，使用者可以自創新模組、命名新功能，再依邏輯思惟將這些模組堆疊、設計指令。

結合翻轉學習模式，在 CS10，每個學生都配備一個控制器，可以在課堂上和教授互動。全身散發著熱情的葛西亞也很習慣戴著耳麥，在偌大的禮堂裡穿梭講課。

那天談的是如何程式中的抽象概念，對他來講，寫程式更多是關於所謂的想法（big ideas），而不是細節，「我要向學生證明，從遊戲學習他們也學得到 big ideas，將有創意的想法，化作真實的產物。」

他口中的 big ideas，指的是程式語言裡面談到的迴圈、抽象化、通則化等概念。

Capturing the Moment of Creativity

In contrast to CS61A, which students believe is overly challenging, CS10 is more geared towards students who are not majoring in computer science, with a focus on fun learning opportunities to spark student interest in programming.

This course pioneered the Snap! program. Unlike other languages that have lines and lines of code, Snap! is a visual language that uses blocks to allow users to create new combinations, name new functions, then logically assemble these pieces to perform commands.

CS10 also flips the classroom model by giving every student an iClicker so they can interact with the professor. During lectures, a wholly enthusiastic Garcia often traverses the length of the auditorium with his mic.

The topic of the day was the abstract concepts behind programming. For him, writing programs is more about the big picture than it is about small details. “I want to show students that they can learn big ideas even through game-based learning, and that this creativity can produce real results.”

The big ideas he speaks of are the ones common to computer programming: for loops, abstraction, generalization, and so on.

雖然讓學生從遊戲中學習，這套軟體卻可以套用在不同的程式語言，透過遊戲學習程式的基礎概念，並運用基礎概念，解構大問題變小問題。

「先讓學生思考，自己想做什麼題目，抓住有創意的動機，再跟他們說需要具備什麼工具，」電機工程與電腦科學系兼任講師弗里蘭（Gerald Friedland）認為，傳統電腦科學教育太強調理論跟數學模型，很多課程的教法甚至都還停留在七〇年代，而學生不知道自己有能力做出產品，甚至覺得跟自己無關，就沒有動機學習。

除了電腦運算思惟之外，這門課讓學生在期末用 Snap! 這套軟體做開放式專案，不指定題目和形式，學生想做什麼都可以，去年學生還結合 Arduino 設計自己的作品。

玩出改變社會的可能性

葛西亞興奮地拿起電腦，就想展示給《天下》記者看各種專案。他如數家珍，有的團隊設計機器人，能透過感測器找到迷宮出口；還有學生畫笑臉，讓電腦辨識追蹤笑臉的線條弧度；更有跳舞機器人會記得人類肢體動作，記錄下來後跳成舞蹈。

Even though Snap! teaches students through games, the concepts behind it are applicable for other languages, and through the foundational ideas, students learn to break large problems into smaller ones.

“Let the students come up with what they want to do, capture that creative motivation, and then tell them what kind of tools they need,” says EECS and Cognitive Science professor Gerald Friedland. According to him, computer science education has traditionally overemphasized theory and mathematical models, and many teaching models have remain unchanged since the 70s. As a result, students have no motivation to learn CS because they don’t know that they have the power to create products, and even believe that CS has nothing to do with them.

Aside from computational thinking, this course also allows students to design their own Snap! projects at the end of the semester. There are no set topics or structure, and students are free to do whatever they’d like. Last year, some students even used Arduinos to design their projects.

Using Play to Unlock the Potential for Societal Change

Garcia grabs his laptop excitedly, eager to shows off all sorts of student projects to the CommonWealth magazine reporter. He’s familiar with each one of them: this team made a robot with sensors to navigate a maze, that team drew a smiley face and had a computer analyze the curvature, that team made a dancing robot that memorizes choreography from human actions...

弗里蘭也舉例，曾有學生直接到辦公室說想做總統大選，想結合民調數據，但因為問題太大，還得找統計系的教授合作。

葛西亞很自豪，這堂課的角色就是作為引子，讓學生不會排斥接觸電腦科學。

「那時候大家都學得很開心啊！」今年柏克萊經濟系畢業的林志學現在在新創公司當產品設計師，他回憶這堂課像在玩積木，兩、三人一組做小遊戲，好玩也學會邏輯思考。

除了讓大家有興趣，還得開拓視野。在這堂課，可以聽業師跟「課」友回來，分享自己如何在專業領域運用過去學到的經驗。

不管是 CS61A 或 CS10，這兩門在柏克萊最受歡迎的程式課，初心就是讓更多學生知道，自己是有能力藉由數位工具和思惟的轉變，看到改變的可能。

「這是第一堂課，不管學生未來會不會從事相關的工作，最重要的是學過這門課就會知道什麼都有可能，」葛西亞這麼說，他的眼神裡有著期待，期待更多學生願意為了改變社會，影響世界而往下深耕。

Original Article:
<http://www.cw.com.tw/article/article.action?id=5079340>

Author: 程晏鈴

Friedland says that one time, students came to office hours wanting to combine various polls for the presidential election. They even recruited a professor from the statistics department to assist them with their monumental task.

Garcia is immensely proud of how the course has become an attractive introduction, so that students no longer shy away from computer science.

Zhixue Lin, a recent Berkeley graduate who studied economics, is now a product designer at a startup. He reminisces about the course, saying that it was like playing with Legos in teams of two or three, building games and learning to think logically. “We had a great time learning!” he says.

Aside from inspiring interest, the course also reveals the breadth of opportunities available for students. Students have a chance to listen to working professionals and course alums speak about how they’re using their skills in their current professions.

Whether it’s CS61A or CS10, the two most popular programming courses at Berkeley were created to let students know that they have the ability to use digital tools and a different way of thinking to enable change.

“This is the first course. No matter what the students do in the future, the most important thing is that after this course, they know that anything is possible,” Garcia says with hope in his eyes, eagerly awaiting more students who are willing to change society and influence the world.

Translated by: Raya Juitung Kuo