

創客教育、運算思維、程式設計~幾個從「想」到「做」的課程與教學設計觀念

劉明洲

國立東華大學教育與潛能開發學系教授

隨著文憑主義的衰退，這一波教育思潮再度風起雲湧地重視給予學生實作的體驗，同時也期待能為未來培養創新的重要種子。因此，從過去單向「想」的學習模式，欠缺「實作」的學校課程，到今日創意創新成為競爭主體的時代來臨，確實逐漸在翻轉教育的觀念。這當中鼓勵學生自己動手做出東西的創客(Maker)教育(亦有稱之為「自造者教育」)、利用電腦解決問題的運算思維(Computational thinking)，以及實際進程式編碼的程式設計(Programming)，三個課程內涵的關係為何?是值得進一步釐清的，尤其在中小學階段，各個議題的適用內涵為何?各自要達成怎樣的目標?甚或之間如何相互支援以達到精緻化的課程教學?本文將簡略做幾點提醒。

創客教育的基本精神，在鼓勵學生將自己的構想或創意，使用雙手或自有機具做出產品。除了鼓勵學生動手做的習慣，進而要落實的是活用知識的習慣與能力，甚至可鼓勵與外界的實體世界連結，鼓勵創新、生產、與創業。時至今日，主要借重的工具就多屬數位工具了。儘管非數位的手作也還是很重要，但能使用電腦設計，結合3D列印零件組裝成品將會是主要的過程，也就是數位創客的型態，會讓創客的歷程更加有效能。推動創客教育要思考的是學生在動手與忙碌之餘，更深刻的學習內涵應該是甚麼?換個問題思考，在創客展上看到許多大中小型的學生作品，與

過去的美勞課或工藝課有何不同?我會認為還是要鼓勵創新、生產、甚至與創業相關，可以讓創客教育的實施更加聚焦，在把動手做變成是習慣與興趣之餘，每個國民都能有從使用者的立場去想到製造者乃至發明者源頭的素養。

運算思維原來頗強調演算過程，尤其運用在程式設計思考上，這在中小學程式設計教學上的啟示是解題腳本(Plan)的構思與驗證可能比程式碼完成否更重要，而程式碼的寫作與解題腳本的印證思考可能比熟悉程式碼本身重要。所以要鼓勵學生做解題思考，也就是用演算法(Algorithm)去把解題歷程規劃出來的部分，甚或可以結合一些創意在其中。而這種思考會與數理學科頗有關聯，卻未必與數理成績有絕對關係。在落實運算思維的教學上，重點還是要落實數理教學，尤其在STEM(Science, Technology, Engineering and Mathematics)課程上的互相連結與整合。除了知識面的學習與應用，情意面對於問題解決的鏗而不捨態度培養，也應該是一個重點。

程式設計過程主要是在讓學習者透過程式語言編碼(Coding)形成可以解決問題的程式，其重要性似在運算思維之下，因為先有解題計畫才能據以產生其解題程式。但其實二者的關係密切，因為沒有程式設計則運算思維會變得空泛，換言之，程式設計讓運算思維更具體化。過去程式語言的

學習比較艱難，而且都從變數宣告、語法熟悉等入門，很容易讓初學者在一開始就打退堂鼓。時至今日，圖形化、物件化的介面(例如 MIT 的 Scratch 語言)讓程式語言的學習變得比較有趣也容易入門。程式語言教學部分要關注的是教學法的問題，尤其工具本身是更適合中小學生學習了，但是教學的部分是否與學習者的認知需求匹配則有很大的努力空間，現場有很多老師是用他自己的學習方法，也就是他的成功學習經驗在教小朋友學習，但其實二者是不一樣的。尤其在把程式設計列入為 107 課綱的政策期待下，2018 年後，國中小學生都很有可能要學程式設計，其中的師資與教材教法都變得更重要，否則，讓人擔心的是會像數學學科在中小學的窘境一般，重要性沒問題但要學得好卻大有問題，甚至成為許多學生頭痛的一科，難保以後程式設計也成為了補習班的重點科目，則對學生而言，是未蒙其利卻先受其害了。

針對程式設計的教學方法，在中小學階段可以把握幾個要領，1)先用學生學習的角度去思考應該如何教學，老師不要自己是怎麼學的就要用同樣方法教學生學習。2)先用簡單的例子或任務讓學生建立成就感。3)由小到大，例如把幾次的小成果組合成一個大成果。4)可以聚焦於部分重點，讓學生做程式的局部改寫，而不必面對太多的程式碼而亂了焦點。5)許多程式語言都有逐步執行追蹤(Trace)或自動偵錯(Debug)的功能，可以善用來讓學生做驗證思考。這些方法的重點都是在減輕學生的認知負荷，才有機會學得好，學的久，學出成就與樂趣來。

創客教育、運算思維、與程式設計這三者之間在教學活動設計上的關係「可有可無」，也就是可以有關係，也可以沒有關係，端視學生的需要與教學的情境是否允許。常在教學現場聽到家長或老師們說「因為要推創客，所以要學程式設計」，其實若學生能力與設備條件允許，並且也能設計出教學活動將三者整合，當然沒有問題，但若有任一條件不允許，則個別實施亦無妨，只要學生學得有興趣都是好事。

重點說明了創客教育、運算思維、與程式設計的實施重點之後，在此提出總體性的課程與教學建議：

1. 釐清各階段的學習目標，據以規劃可行的教學內容：小學、中學、大學一定會有所區隔，例如小學階段著重在發想與體驗；中學可以更著重實用性的作品製作，大學生則可以具體的導入生產與創業的相關連結。如此區隔雖非絕對的涇渭分明，但可以做為整體的參照。於是，教師可以在教學引導上增加彈性，例如有些有意義卻艱難的內容或許給小學生體驗即可，或是打包成黑箱(Black box)讓小學生了解其功能卻未必進入其複雜的內容，讓整個學習歷程不因此受阻。
2. 透過與其他課程的連結讓教學更順遂：因為創客教育、運算思維、與程式設計都不應該被視為是全新出現的課程，而是在我們的生活與學習脈絡下的一個自然應運而出的課程，因此在課程與教學設計上可以多與生活脈絡結合，也多與其他課程連結，讓學生的學習與成長更具統合性與一致性。

3. 創客教育、運算思維、與程式設計除了要培養能力，但興趣與習慣還是遠重要於知識的學習，因此，學習的氛圍應該是自由的、愉快的、被鼓勵的、具有挑戰的。應該要避免太有壓力的任務目標考核或是

學習成就評量，以避免揠苗助長。培育人才需要時間，也需要耐心。家長或校方常常因為期待過高，或著急於達成目標，反而扼殺了學生的創意與動力，這才是最划不來的教育投資。

