

計算思維—編程教育

小學課程補充文件

課程發展議會編訂

香港特別行政區政府教育局建議學校採用

二零一七年十一月擬訂稿

如對本課程文件有任何意見和建議，請致函：

香港九龍塘沙福道19號
教育局九龍塘教育服務中心西座1樓W101室
教育局課程發展處
總課程發展主任（科技教育）收
傳真：2768 8664
電郵：teched@edb.gov.hk

目 錄

第一章	引言	4
第二章	目標	5
第三章	學習元素	6
第四章	實施	7
	課節安排	7
第五章	學與教	8
	主導原則	8
	計算思維和編程學與教的一些注意事項	9
第六章	評估	10
	主導原則	10
	評估計算思維實踐的發展	10
附錄		12
	附錄一 電腦認知單元課程教學安排建議	12
	附錄二 時間分配建議 (第二學習階段適用)	13
	附錄三 評估工具	21
	附錄四 詞彙釋義	23

第一章 引言

於 2016 年 12 月發布的《推動 STEM 教育－發揮創意潛能》報告，強調加強學生綜合和應用不同科學、科技、工程和數學(STEM) 學科知識與技能的能力，培養他們的創造力、協作和解決問題能力，以及促進發展二十一世紀所需的創新思維和開拓與創新精神，並提供了在中小學推廣 STEM 教育的具體例子，包括建議在小學階段引入編程¹來培養學生的計算思維，希望通過適當設計的學習活動，為學生提供獲取和應用計算思維²和編程技巧的機會。

進入資訊時代，互聯網的擴散和資訊及通訊科技的進步，為社會和個人帶來了巨大的影響和轉變。今天，社會上有強烈需求，希望裝備我們的學生，能擁有藉編程來處理信息和計算設備的知識和技能，以滿足這個互聯世界日益增長的挑戰。

在小學階段，常識科為學生提供了在個人、社會及人文教育、科學教育，以及科技教育學習領域中使用資訊科技整合知識、技能、價值觀和態度的機會。課題包括透過手腦並用的學習經驗和活動來培養學生的解決問題能力。此外，自 2000 年以來，教育局已開發了八個電腦認知單元課程 (CAP)，讓學校使用，當中包括基礎編程單元，以助學生掌握基本的編程技巧。學校多已運用有關資源，並於每級開設每星期一課節的校本資訊科技課程，其中包括教授編程。這些單元課程已於 2015 年修訂和更新，**附錄一**提供有關單元一至七的一些教學安排建議，以供學校參考。

編程教育的重點是讓學生掌握編程的技巧，並且將編程技巧應用到不同的情境（電腦系統-軟件）之中，以完成所設定的任務/工作。在考慮現有的學校實際情況，以及展望互聯世界的最新發展，本補充文件旨在為計算思維和編程教育提供一個有系統及詳細的描述，並列述相關的學習重點，以便教師在小四至小六施教，培養學生必要的知識、技能和態度，以面對未來日益增長的數碼經濟時代。一般而言，電腦認知單元課程及常識科課程中有關資訊科技的部分已經涵蓋大部分編程教育學習內容。本補充文件提供了一個綜合而全面的說明，以助學校有系統地在課堂上教導相關課題。

計算思維和編程是一種可以在不同學習階段，以及各種學科中使用的解決問題方法。學校可以在第一學習階段提供一些日常生活體驗，如遊戲、體育和音樂，以助啟蒙學生日後在計算思維和編程的學習。其中不少遊戲活動，本身都有其規則及步驟，當中涉及次序及勝負條件，已是計算思維的初步概念，初小學生可以從生活體驗中接觸相關的概念。

¹ 在文件中，「程式編寫」（或簡稱編程）和編碼的意義上是相同的，廣義上是指把一個原始方案問題（或稱計算問題），經過一個過程，而引出一個可執行的程式（或稱電腦程式）。

² 周以真教授於 2006 年提出了「計算思維」的概念，指出「計算思維是運用計算機科學的基礎概念去解決問題、設計系統和理解人類的行為。」（Wing, 2006）。周教授還強調：「計算思維是每個人的基本技能，不僅僅屬於計算機科學家」。

第二章 目標

完成後，學生將能夠：

- 瞭解計算思維的基本概念，包括抽象化、算法和自動化。
- 將編程與現實生活中的問題和其他科目連繫起來。
- 具備有開發／重用／混合程序／編碼以解決基本計算問題的能力。
- 瞭解編程的局限性和影響。
- 在過程中溝通及有效的團隊合作以解決問題。

在高小年級推行計算思維和編程教育，目的並非訓練及培養電腦程序編寫員，而是讓學生得到實作經驗及建立解難的信心，持續透過協作及重覆的測試來解決問題。

第三章 學習元素

學習元素內容重點在發展高小學生計算思維和相關編程的技巧。我們期望學生學習以下內容：

- **計算的連繫**
 - 硬件和軟件作為一個系統以完成任務
 - 辨別使用電腦帶來的影響
 - 描述人與計算之間的連繫和發展
 - ◆ 產消合一（生產和消費的模糊化）
 - ◆ 個人化建議
 - 計算的限制
 - ◆ 過於複雜的議題（例如：天氣預測）
 - ◆ 不可能計算
- **計算思維的實踐**
 - 識別數據及計算使用的資訊，以及如何使用數據
 - ◆ 事實陳述（數據類型和詳細程度）
 - ◆ 「如何做」的知識
 - 製造計算製品
 - ◆ 儲存程式電腦的概念
 - ◆ 使用變量以便儲存及修改數據
 - ◆ 基本程序編寫的結構
 - 序列
 - 分支／選擇
 - 循環
 - ◆ 抽象的概念
 - 格局圖樣
 - 模組化
 - ◆ 開發程序／編碼
 - 程序編寫語言的語法和語義
 - 設計、重用、混合程序／編碼
 - 搜尋／排序算法
 - 使用綜合開發環境 (IDE)
 - ◆ 與實物進行互動
- **應用和影響**
 - 連接到其他部件形成一個互相依賴的系統（例如：智能家居系統）
 - 網絡安全、私隱和保安措施
 - 數據累積和準確性
 - 溝通、協作和創新，以滿足人們的需求和希望

第四章 實施

我們建議學校可按學生需要和學習階段的特有情況，靈活調配課時進行相關的學與教，例如利用課程內「可供彈性處理的時間」，在高小年級每級安排每星期一課節（每課節 35／40 分鐘），以涵蓋本補充文件的相關內容及 CAP 內涵蓋的其他資訊科技的技能和知識。

建議的學習重點旨在培養學生計算思維。然而，計算思維的學與教可能已經在現時學校的課堂內進行，它與學校 STEM 學習中的專題研習可緊密地連繫。

課節安排

現時學校大多已在每級每星期撥出一課節開設「校本資訊科技課程」，教授電腦認知單元或相關內容。學校可按照個別的情況靈活調配課時實施編程教育。在此，我們提供了兩種實施模式供學校參考。

- **以學校為本位課程涵蓋高小的計算思維和 CAP 教學：**建議學校每週在每個級別分配一課節（每課節 35／40 分鐘），以涵蓋相關內容及 CAP 內涵蓋的其他資訊科技的技能和知識。課程總課時約每級每年為 18 小時（約 32 課節）。學校可以靈活設計自己的實施時間表，例如「時段編課方式」，只在一學期每週提供一個連堂的雙課節，而不是每週一課節。這種安排具有較長的課堂時間，以及減少堂與堂之間的過渡時間和其他課堂管理活動，因此學生可以有更多時間來完成任務、練習和學習教學內容。
- **在高小的個別科目以主題式教學：**學校若認為合適，可以考慮將相關內容併入高小的個別科目，如數學科、常識科的教學中。學校可以考慮在高小年級每個級別的相關科目中每週調撥多一課節（每課節 35／40 分鐘），以教授相關內容和 CAP 內涵蓋的其他資訊科技的技能和知識。學校可採用主題式教學，把學習重點集中在一個主題上涵蓋一系列內容和技能。這種教學策略將不同的內容連繫，創造一種學習目標。學生通過利用自己的興趣和生活經驗，知識、技能和態度便能以更有意義的方式發展起來。

為了使學生能夠練習他們的編程技巧並豐富他們的程序編寫知識，建議學校每學年在高小分配 8 至 16 課節，在電腦輔助學習室或藉移動設備在課室內進行編程練習。

我們鼓勵教師靈活運用時間，為學生提供課堂內外的各種學習經驗，以達致計算思維的學習重點。目前，頗多學校正採用校本課程，安排每星期每級一課節引入電腦認知單元課程。建基於學校現時的做法，**附錄二**是有關高小計算思維與電腦認知單元課程的校本課程時間分配建議，以供學校參考。

第五章 學與教

計算思維和編程的學習是一個複雜、多元化、生動而互動的過程。除了傳統的講授方式外，教師應將主動學習的元素融入課堂活動中。教師應運用多樣化的教學策略，指導學生如何利用多種方法，以獲取本補充文件所涵蓋的知識、概念及能力，例如解決問題、明辨性思考、創意、溝通、後設認知及提升學會學習的能力。負責計算思維和編程的教師除對計算思維和編程有一定的認識外，亦須掌握相關的教學方法、知識和技巧。

主導原則

以下部分概述基本學與教理念及其主導原則：

- **知識**：知識具不同形式與情境。某些知識是學生已經建立的，另一些則是靈動多變及具情境的。有用的知識須由學習者主動建構而成。
- **學習**：學習可透過不同途徑進行。知識可透過教師指導及閱讀文獻獲取，也可通過學生自我反思及與他人協作互動等不同活動來獲取。
- **清晰的學習目標**：每項學習活動的設計，對教師和學生來說，都應該具有清晰明確的學習目標。
- **促進理解的教學**：教學法的採用，目標是令學生明白他們在學甚麼，而不是只求強記內容。
- **建基於已有知識和經驗**：計畫進行某個學習活動前，教師應先充分掌握學生的已有知識和經驗。
- **善用一系列的教學法**：設計多樣的學與教方法和活動，以配合不同的學習目標及學生學習模式，從而達至有效的學習。
- **促進互動**：教師善用互動以帶領學生嘗試不同的意念，從而探索甚麼是學生知道的、甚麼是不知道的。教師應運用開放式問題使學生思考及提出自己的意見，令學生能互相學習。
- **促進獨立學習**：透過課程內容的學習活動，培養學生的共通能力及反思能力。教師應鼓勵學生為自己的學習負起主動而認真的責任。
- **善用進展性評估**：設計評估活動，搜集和提供資訊，通過善用進展性評估有效地改進學與教。
- **有效運用資源**：運用不同種類的教學資源作為學習的工具。
- **增強動機**：是否達至學習目的，往往取決於學生有沒有較強的學習動機。教師應使用適當的誘發動機策略，從而喚起學生的學習興趣。

- **致力參與**：在進行學習活動時，教師應使所有學生均可積極參與，並致力確保每個學生在活動時都能致力參與。
- **照顧學習者多樣性**：學習者的特點及能力各有不同，教師應運用各種策略，以照顧學習者的多樣性。例如，建立一個學習社群，讓不同能力的學習者可在社群內互相為對方的學習作出支援。

計算思維和編程學與教的一些注意事項

- 就已知和不明問題都分別進行練習
- 投入時間培養分解問題的技巧（即將問題分解成更易於管理和易於理解的較小部分），以及在學習程序編寫技巧前先學習識別數據和信息以作計算的用途
- 投入時間通過不插電活動來發展抽象化的技能
- 有創意及不要害怕失敗

第六章 評估

主導原則

評估的目的是要收集學生的學習顯證，讓各持份者包括學生、教師、學校及家長等瞭解學生的學習進度。

在發展與科技教育學習領域內容有關的評估的策略時，學校可留意下列主導原則：

- 科技教育是一個有目的而全面的學習，評估應配合有關的學習元素及能夠反映其重要學習部分，包括知識、概念、過程、覺知、共通能力、價值觀和態度。
- 評估目的是確保學生能夠獲得足夠的知識和技能，並能平穩進步。尤為重要的是，在科技學習中，學生能夠遵守安全和健康守則來使用工具和儀器。
- 評估必需融入學習過程之中。一般而言，進展性評估(即回饋學與教評估)和總結性評估(即評估所學)，在提高學生學習能力和描述學生的學習進度這兩方面，都是同樣重要的。計算思維學習是為所有能力的學生而設，旨在為學生提供使用電腦／工具來加強他們獲得相關概念的機遇。因此，應使用如課堂參與、家課和專題習作等進展性評估。這些以表現為主導的評估方式，能為學校提供重要的資料，以便教師不斷進行教學檢討及作出回饋，從而協助學生求取進步。
- 尤其值得重視的是，要讓每一位參與評估者，包括教師、學生、家長等，都理解和懂得運用評估結果，並能根據評估結果，制定未來的學習方向。

評估目標必須與之前章節所表述的學習元素相配合，期望學生能夠：

- 把一個簡單問題抽象化以便設計一個解決方案
- 了解基本的程序編寫結構，並能在日常生活中看到相關的結構
- 預測更改程序語句／代碼如何能夠改變程序的操作／輸出
- 熟悉重用和混合程序／編碼，並能夠測試和修復程序中的錯誤
- 找出程序中的格局圖樣，並能夠在新的情況下應用它們
- 利用程序編寫語言表達一個簡單的算法

評估計算思維實踐的發展

*附錄三*是一評估工具，改編自哈佛大學教育研究院 ScratchEd 小組開發的“Creative Computing”小冊子內的評估工具。有關工具詳情可參考網址：<http://scratched.gse.harvard.edu/guide/>。

教師可以參考這工具來評估學生透過編程發展計算思維實踐的能力。在學期內，教師應持續觀察及記錄學生的行為，作為參考基礎，在學期完結時給予學生一個應得的學業成績。

參考

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Brennan K., Balch C., and Chung M. (2014). **Creative Computing**, US : Harvard Graduate School of Education

附錄

附錄一 電腦認知單元課程教學安排建議

單元名稱	註釋	
	第一學習階段	第二學習階段
(一)開開心心用電腦	<ul style="list-style-type: none"> ● 使用鍵盤及滑鼠控制電腦 ● 熟習在視窗環境中操作電腦系統 	<ul style="list-style-type: none"> ● 認識日常生活中需要資訊科技的原因及其應用 ● 明白輸入、輸出及儲存裝置的基本功能 ● 簡單地認識電腦如何運作 ● 遵守使用電腦的守則
(二)用電腦繪圖	<ul style="list-style-type: none"> ● 利用手畫及橡皮擦工具繪圖 ● 利用基本繪圖工具編輯圖片 ● 在圖片中輸入文字 	<ul style="list-style-type: none"> ● 運用一般繪圖工具編輯、儲存及列印圖像檔案
(三)用電腦書寫	<ul style="list-style-type: none"> ● 熟習鍵盤中各個鍵的排列，並學習鍵盤操作技巧 ● 利用手寫識別裝置，輸入中文字 ● 利用文字處理軟件的功能編輯、格式化、儲存及讀取文件 	
(四)使用互聯網		<ul style="list-style-type: none"> ● 初步認識互聯網和電子郵件的功能和應用 ● 利用瀏覽器瀏覽互聯網及搜集關於特定課題的資料 ● 關注使用互聯網時的法律知識及培養正確使用互聯網的態度
(五)文字處理		<ul style="list-style-type: none"> ● 學習文字處理軟件的功能 ● 認識不同的中文輸入法，並能透過鍵盤輸入中文字
(六)試算表計算及圖表製作		<ul style="list-style-type: none"> ● 認識在日常生活中使用試算表的需要 ● 以表格形式或圖表方式表達數據資料 ● 利用公式處理簡單的計算，例如加數及平均數
(七)使用電子郵件		<ul style="list-style-type: none"> ● 認識電子郵件的操作 ● 培養正確使用電子郵件的態度 ● 認識其他電子通訊的方法，例如網上交談和視像會議

附錄二 時間分配建議 (第二學習階段適用)

以下時間分配 (以 35/40 分鐘的課節表示)只供參考，學校可因應校本需要作出調整。

	建議課節數目			
	小四	小五	小六	
電腦認知單元課程				
(一)開開心心用電腦	2			<ul style="list-style-type: none"> ● 認識日常生活中需要資訊科技的原因及其應用 ● 明白輸入、輸出及儲存裝置的基本功能 ● 簡單地認識電腦如何運作 ● 遵守使用電腦的守則
(二)用電腦繪圖	4			<ul style="list-style-type: none"> ● 運用一般繪圖工具編輯、儲存及列印圖像檔案
(四)使用互聯網	4	4	2	<ul style="list-style-type: none"> ● 初步認識互聯網和電子郵件的功能和應用 ● 利用瀏覽器瀏覽互聯網及搜集關於特定課題的資料 ● 關注使用互聯網時的法律知識及培養正確使用互聯網的態度
(五)文字處理		2	2	<ul style="list-style-type: none"> ● 學習文字處理軟件的功能 ● 認識不同的中文輸入法，並能透過鍵盤輸入中文字
(六)試算表計算及圖表製作		6	6	<ul style="list-style-type: none"> ● 認識在日常生活中使用試算表的需要 ● 以表格形式或圖表方式表達數據資料 ● 利用公式處理簡單的計算，例如加數及平均數
(七)使用電子郵件	4			<ul style="list-style-type: none"> ● 認識電子郵件的操作 ● 培養正確使用電子郵件的態度 ● 認識其他電子通訊的方法，例如網上交談和視像會議
小計	14	12	10	

	建議課節數目			
	小四	小五	小六	
計算思維－編程教育				
計算的連繫				
<ul style="list-style-type: none"> ● 硬件和軟件作為一個系統以完成任務 	0.5			<ul style="list-style-type: none"> ● 識別和探索數碼系統（硬件和軟件組件）的用途，例如裝有微控制器的洗衣機、數碼相機、智能手機
<ul style="list-style-type: none"> ● 辨別使用電腦帶來的影響 	0.5			<ul style="list-style-type: none"> ● 意識到微處理器和圖形用戶界面(GUI)的發展便利了人機互動(HCI) ● 意識到網絡和互聯網的影響，帶來地理資訊地圖(GeoInfo Map)、對等(P2P)和相關系統的發展
<ul style="list-style-type: none"> ● 描述人與計算之間的連繫和發展 <ul style="list-style-type: none"> ■ 產消合一（生產和消費的模糊化） ■ 個人化建議 	0.5	1		<ul style="list-style-type: none"> ● 意識到人們收集數據和使用它們以便管理日常操作 ● 意識到互聯網所引致新議題的出現，例如全球化和開放式社會，以及令隱私減少 ● 意識到「產消合一」的興起，其中涉及生產和消費的共融 ● 意識到網絡上「用戶生成內容」的爆炸性發展，提供不少免費產品、和引致沒有報酬工作的出現。例如在社交網站中，用戶自行創建用戶資料，其中包含了一些視頻、照片、和文本，與他人互動並建立社群 ● 意識到社會上勞動力的減少，人們會更多「自助掃描」或「移動掃描式地購物」，例如自助檢查終端機的興起 ● 意識到供應商建基於客戶在網絡上過去的行為作風，為客戶們提供貼身和相關產品／服務的建議 ● 意識到供應商收集和使用網絡上客戶的數據，可為客戶們在瀏覽網絡上的一刻提供適時和相關的產品／服務
<ul style="list-style-type: none"> ● 計算的限制 <ul style="list-style-type: none"> ■ 過於複雜的議題（例如：天氣預測） 	0.5	1		<ul style="list-style-type: none"> ● 意識到電腦在速度、記憶體、以及精確度方面都比人優勝 ● 意識到電腦要被編程才能操作。一些太複雜問題是無法開發（如天氣預報）或無法計算（如人的感覺）

	建議課節數目			
	小四	小五	小六	
<ul style="list-style-type: none"> ■ 不可能計算 				<ul style="list-style-type: none"> ● 意識到信息儲存和通訊等的限制 ● 意識到如未能精確地知道某些情況，會導致系統內一些不確定的行為／操作的出現；意識到如未能完全地隔絕，系統會不可避免地與一些未知環境產生互動 ● 意識到人類的思想是不可能被模仿，由於： <ul style="list-style-type: none"> ■ 思維是通過互動和動腦筋地顯現出來 ■ 思維不是一個存儲在某個地方的數據 ■ 思維不能完全被客觀化、孤立、縮減、或重複 ■ 人類樂意的感覺、夢等不能被衍生
計算思維的實踐				
<ul style="list-style-type: none"> ● 識別數據及計算使用的資訊，以及如何使用數據 <ul style="list-style-type: none"> ■ 事實陳述（數據類型和詳細程度） ■ 「如何做」知識 	1	1		<ul style="list-style-type: none"> ● 以考試/測驗，或超市為議題，瞭解和： <ul style="list-style-type: none"> ■ 指出議題中包含什麼項目 ■ 識別項目的詳細程度 ■ 識別數據 ■ 認識和探索數據中的格局圖樣 ● 使用考試/測驗中的得分、分數和等級為例，瞭解和找出考試或測驗後的等級 ● 以超市的結帳排隊為例，意識可能影響選擇排何隊列的參數，例如：等候隊伍的長度、裝袋工的安排、是否快線、客購物車中物品的數量
<ul style="list-style-type: none"> ● 製造計算製品 <ul style="list-style-type: none"> ■ 儲存程式電腦的概念 ■ 使用變量以便儲存及修改數據 	1	1	2	<ul style="list-style-type: none"> ● 意識到器具內儲存了指令，以便這器具能夠按序執行各種任務，例如，電飯煲、科學計算機、筆記簿型電腦、平板電腦 ● 意識到使用變量，作為電腦內的容器，以便保留在程式中被引用和操作的資訊

	建議課節數目			
	小四	小五	小六	
				<ul style="list-style-type: none"> ● 意識和明白不同類型的變量，如數值變量和字符串變量
<ul style="list-style-type: none"> ◆ 基本程序編寫的結構 <ul style="list-style-type: none"> ● 序列 	2	1	2	<ul style="list-style-type: none"> ● 意識「序列」為設計算法中一個基本結構（組件） ● 明白「序列」中的每個步驟為一個指令，並在算法中按照次序逐一執行。例如：清潔牙齒的算法 ● 意識及明白任務次序中，正確指令或步驟必需依一正確序列，例如綁鞋帶、按照食譜烹調、使用計算機計算「$a + b \times c$」算式、按照課業與考試的分數，計算已經加權後的分數 ● 意識有些次序並非重要，如寫郵寄地址的「常規次序」
<ul style="list-style-type: none"> ● 分支／選擇 				<ul style="list-style-type: none"> ● 意識「分支／選擇」為設計算法中一個基本結構（組件） ● 明白在算法中的某一步驟可能出現多於一個方法／路徑來處理，而發生的位置與時間需要作出「選擇」或「決策」，例如支付巴士車資 ● 意識及明白單一分支 IF（如果下雨我便使用傘子）及多個 IF（找出某一月份的日數）算法所涉及的步驟。例如識別及尋找康文署游泳池等的開放時間
<ul style="list-style-type: none"> ● 循環 				<ul style="list-style-type: none"> ● 意識「循環」為設計算法中一個基本結構（組件） ● 明白「循環」中的某些步驟需要重覆執行直至被命定或在特定條件成立下停止 ● 意識「循環」的重要性，能令電腦進行重覆性的工作。「循環」有助簡化算法。例如：清潔牙齒的算法 ● 意識需要設定重覆的步驟及停止的條件。例如於圖書館尋找某本書，晚餐後清洗碗碟。認識「循環」的不同停止條件，例如，計數器控制、警戒值控制、與無終止
<ul style="list-style-type: none"> ◆ 抽象的概念 	2	1	1	<ul style="list-style-type: none"> ● 意識「抽象」為找出事物以便為問題製造一模型的過程

	建議課節數目			
	小四	小五	小六	
<ul style="list-style-type: none"> ● 格局圖樣 ● 模組化 				<ul style="list-style-type: none"> ● 能夠識別事物的屬性、功能、以及其介面（即事物間的通訊辦法） ● 明白忽略一些不需要圖樣特徵的重要性，以便能夠將事物減少到最基本，以一些必需的元素為代表。例如能在以下數字序列中找出 n：3, 6, 9, 12, 15, n 及 3, 3, 5, 4, 4, 3, 5, 5, 4, n ● 意識格局圖樣的出現。例如利用方格紙繪畫下列次序的長度線條：1, 3, 5, 7, 9, 1, 3, 5, 7, 9, ...，每畫完一線段後順時鐘轉 90 度 ● 意識及明白「分治法」作為一個問題解決的方法，將一個大問題切分成幾個較小的程序來解決問題 ● 識別一些可重用的程式碼，而非連續性地鍵入一些相同的語句 ● 意識按照「功能性」把問題切分成為小問題，並逐一解決的需要。例如繪圖，通過點繪製正方形、多邊形、圓形；通過三維空間以點繪製正立方體、球體 ● 意識模塊介面的使用，例如模塊的命名，輸入和回送的數據 ● 意識模塊之間、以及與主程序之間的執行流程
<ul style="list-style-type: none"> ◆ 開發程序／編碼 <ul style="list-style-type: none"> ● 程序編寫語言的語法和語義 ● 設計、重用、混合 	8	10	10	<ul style="list-style-type: none"> ● 能夠於電腦輸入簡單的程序段來 <ul style="list-style-type: none"> ■ 作數學運算 ■ 寫入句子 ■ 繪畫多邊形 ■ 列印輸出 ● 通過修改簡單的程序段，能夠 <ul style="list-style-type: none"> ■ 識別語法和語義錯誤 ■ 建構和使用測試數據來檢測邏輯錯誤 ● 能夠修改簡單的程序段作

	建議課節數目			
	小四	小五	小六	
<p>程序／編碼</p> <ul style="list-style-type: none"> ● 搜尋／排序算法 [*學生不需在綜合開發環境自行開發個別有關程序。] ● 使用綜合開發環境 				<ul style="list-style-type: none"> ■ 作數學運算 ■ 寫入句子 ■ 繪畫多邊形 ● 能夠在數據上使用適當的運算符、簡單的輸入/輸出、和適當的數據來開發程序 ● 對於給定的任務，能夠在程序上開發和重用適當的分支/選擇結構和循環結構來滿足其要求 ● 意識及使用線性搜尋算法*，按次序逐一檢查序列物件中的每個元素，直至找到目標值。例如從一副撲克牌中找出 Ace 牌 ● 意識及使用冒泡排序算法*把一系列物件排序。例如把一副撲克牌排序 ● 能夠在生活例子上使用搜尋和排序算法，例如協助圖書管理員在學校圖書館將 10 本歸還的圖書放回書架上 ● 能夠進入及離開綜合開發環境並鍵入和儲存程序
<ul style="list-style-type: none"> ◆ 與實物進行互動 		2	6	<ul style="list-style-type: none"> ● 能夠利用綜合開發環境來控制一實物，進行互動和回應一些控制語句
應用和影響				
<ul style="list-style-type: none"> ● 連接到其他部件形成一個互相依賴的系統（例如：智能家居系統） 	0.5	0.5	1	<ul style="list-style-type: none"> ● 認識可以互相共享數據的簡單電腦系統。簡單電腦系統的運作一般是均勻的、線性的、靜態的、和獨立的，例如洗衣機/乾衣機、烤爐、雪櫃 ● 認識近年低成本感應器及高效能嵌入式系統硬件的發展，以及廣泛設立的通訊網絡，使系統中的相互依賴部分容易些管理 ● 認識一些通過關係網絡連接，具有互動和互相依賴部分的複雜電腦系統。意識到系統中存在移動和變化（互動）、以及系統中的部分

	建議課節數目			
	小四	小五	小六	
				發生變化將影響其他部分（互相依賴），例如智能家居系統使用無線網絡（如 WiFi），進行遠程監控家中的照明、通風（如空調）、保安、以及家用電器如洗衣機/乾衣機、烤爐、或雪櫃等的控制及運作
● 網絡安全、私隱和保安措施	0.5	0.5		<ul style="list-style-type: none"> ● 意識到信息共享是網絡安全中一個重要的議題 ● 意識到每個電腦網絡和／或通過互聯網通訊的設備必有一個獨一的互聯網地址－「網絡協定位址（IP 位址）」 ● 意識到私隱涉及建立協議和有效監督個人、公司、或政府部門，在何時、為何、以及如何取得已收集、保留、曾使用、或共享的個人信息 ● 通過以下方式意識和識別保安措施： <ul style="list-style-type: none"> ■ 辨認出網絡上的設備和連接、以及電腦系統和其他設備之間的分界 ■ 執行控制以防止未經授權的存取和檢索、以及濫用
● 數據累積和準確性	0.5	0.5		<ul style="list-style-type: none"> ● 意識到在互聯網上的任何通訊都會留下可以自動化地被收集、保存、和分析的「數字痕跡」 ● 一些組織／公司／人士可能有興趣從數據中創作某個別人士的個人資料。意識到這些「數字痕跡」可以未經用戶的同意、以及用戶的認識從而被收集，保存，發送和處理
● 溝通、協作和創新，以滿足人們的需求和希望	0.5	0.5		<ul style="list-style-type: none"> ● 意識到每個人都是作為一個主體地進行溝通、互相影響、及更有成效地工作 ● 理解到人們通過廣泛媒體分享資訊，從而增加了資訊的流通性 ● 意識到在網絡上需要平衡隱私的價值和資訊自由流通的價值 ● 意識到在編程過程中會自然地產生協作；學生自己在課堂內尋求同

	建議課節數目			
	小四	小五	小六	
				伴的建議和幫助，以及與課／校外人士創立學習社群 ● 通過創造一個能正常運作的程序，產生滿足感和建立信心
小計	18	20	22	
總計	32	32	32	

附錄三 評估工具³

抽象	低	中	高
學生如何決定她／他的程序（專案）需要哪些程序語句（角色），以及他們應該向哪方向處理？	學生沒有提供他／她選擇程序語句（角色）的描述。	學生提供決定選擇某些程序語句（角色）的一般描述。	學生根據程序（專案）的目標提供了她／他如何決定程序語句（角色）的具體描述。
學生如何決定她／他的程序（專案）需要哪些程序段（腳本），以及他們應該做什麼？	學生沒有提供她／他創建程序段（腳本）的描述。	學生提供了創建某程序段（腳本）的決定的一般描述。	學生根據程序（專案）的目標提供了她／他如何根據程序段（腳本）做出決定的具體描述。
學生如何按照對她／他和其他人有意義的方式組織程序段（腳本）？	學生沒有描述她／他如何組織程序段（腳本）。	學生提供了她／他如何組織程序段（腳本）的一般描述。	學生提供了她／他為什麼及如何組織程序段（腳本）的具體示例。

重用及混合	低	中	高
描述學生是否／如何通過嘗試使用其他程序（專案）和閱讀程序段（腳本）來獲得靈感。	學生沒有描述她／他從其他程序（專案）中獲得的想法或靈感。	學生提供了啟發她／她的程序（專案）的一般描述。	學生提供如何啟發她／她的程序（專案）的具體例子。
學生如何選擇另一個程序（專案），並改編於她／他的程序（專案）？	學生沒有描述她／他如何由其他程序（專案）中改編其程序段（腳本）、構想或資源。	學生識別她／他由其他程序（專案）中改編其程序段（腳本）、構想或資源。	學生提供她／他如何由其他程序（專案）中改編其程序段（腳本）、構想或資源的具體示例。
學生如何修改現有的程序（專案）以達到改進，或提	學生沒有描述修改另一個程序（專	學生提供了她／他對另一個程序（專	學生提供她／他為什麼和對其他程序

³改編自哈佛大學教育研究院 ScratchEd 小組開發的“Creative Computing”小冊子內的評估工具。有關工具詳情可參考網址：<http://scratched.gse.harvard.edu/guide/>。

重用及混合	低	中	高
升它的效果？	案)。	案)所作修改的一般描述。	(專案)所作修改的具體例子。
學生如何鳴謝哪些啟發她／他或建基於其作品的人？	學生沒有給別人鳴謝。	學生指出令他／她得到靈感的人的名稱。	學生在程序(專案)和／或網站(如Scratch 網站)中紀錄啟發他／她的人。

測試及除錯	低	中	高
描述當學生運行／他的程序(專案)時發生了與她／他想要的不同之處。	學生沒有描述當她／他運程序(專案)時想要的與得到的有什麼不同。	學生描述了程序(專案)出了什麼問題，但並非他想要做些什麼。	學生給出了一個具體的例子，當他／他運程序(專案)時，發生了什麼事和他／他想要發生什麼。
描述學生如何閱讀程序段(腳本)來找出問題的成因。	學生沒有描述問題。	學生描述了程序段(腳本)，但沒有具體示例找出編程中的問題。	學生描述了程序段(腳本)，並提供了在編程中發現問題的具體示例。
描述學生作出何種改變和測試去看看發生了什麼事情。	學生沒有描述她／他有什麼問題或其解決方案。	學生提供了一個更改程序的一般示例，並測試它是否有效。	學生提供了一個更改程序的具體示例，並測試它是否有效。
描述學生如何考慮其他方式來解決問題。	學生不提供解決問題的示例。	學生提供了一個解決問題的一般示例。	學生提供了一個解決問題的具體示例。

附錄四 詞彙釋義

用語	解釋	頁數
計算思維	「計算思維是運用計算機科學的基礎概念去解決問題、設計系統和理解人類的行為。」(Wing, 2006) ⁴ 。透過運用一些計算思維概念，例如抽象化、算法和自動化，學生將成為工具創造者，而非工具使用者。計算思維是一個可以轉移和應用在不同主題的解決問題的方法。計算思維者會準確描述問題，並構想可以解決問題的算法。	1, 3-10, 14-15
電腦認知單元課程	教育局早於 2000 年已為小學編製電腦認知單元課程，幫助學生掌握電腦知識和基本編程技巧。2015 年更新及豐富了有關單元內容，特別是加入最新的工具（如 Scratch 和 App Inventor）來提升編程的學與教。	3-4, 7, 12-13
編碼	在文件中，「編碼」和「程式編寫（或簡稱編程）」的意義上是相同的，廣義上是指把一個方案問題（或稱計算問題），經過一個過程，而引出一個可執行的程式（或稱電腦程式）。	4-6, 10, 17-18

⁴ Wing, J. M. (2006). Computational Thinking. *Communication of the ACM* 9(3), 33-35.