

運用電子學習工具以提 升學生的計算性思維

伯裘書院
歐海健老師
樂善堂梁銻琚學校(分校)
張家螢老師及蕭燕唐老師



運算性思維

- Computational Thinking Concepts
運算思維知識
- Computational Thinking Practices
運算思維技能
- Computational Thinking Perspectives
運算思維視野



Computational Thinking Concepts

運算思維知識

- 1. Sequences 序列
- 2. Events 事件
- 3. Repetition 重複
- 4. Conditionals 條件
- 5. Parallelism 同步發生
- 6. Naming / variables 命名 / 變數
- 7. Operators 運算子
- 8. Manipulation of data and elementary data structure 數據操作及基本數據結構



Computational Thinking Practices 運 算思維技能

- 1. Reusing and remixing 重用及整合
- 2. Being incremental and iterative 反覆及漸進編程
- 3. Abstracting and modularizing 概念化及模組化 (將想法講出來；畫出來；表達出來)
- 4. Testing and debugging 測試及除錯
- 5. Algorithmic thinking 算法思維 (將想法仔細地表達出來；例：問題仔細講出來；將事拆細(如分成兩部分)來處理→模塊)



Computational Thinking Perspectives

運算思維視野

- 1. Expressing 自我表達
- 2. Connecting 與生活聯繫
- 3. Questioning 提問與了解
- 4. Computational identity 運算身份認同
- 5. Digital empowerment 數碼充權



Framework (Professor KONG Siu Cheung, 2016)

Computational Thinking Concepts 運算思維知識

1. Sequences 序列
2. Events 事件
3. Repetition 重複
4. Conditionals 條件

5. Parallelism 同步發生
6. Naming / variables 命名 / 變數
7. Operators 運算子
8. Manipulation of data and elementary data structure 數據操作及基本數據結構

Computational Thinking Practices 運算思維技能

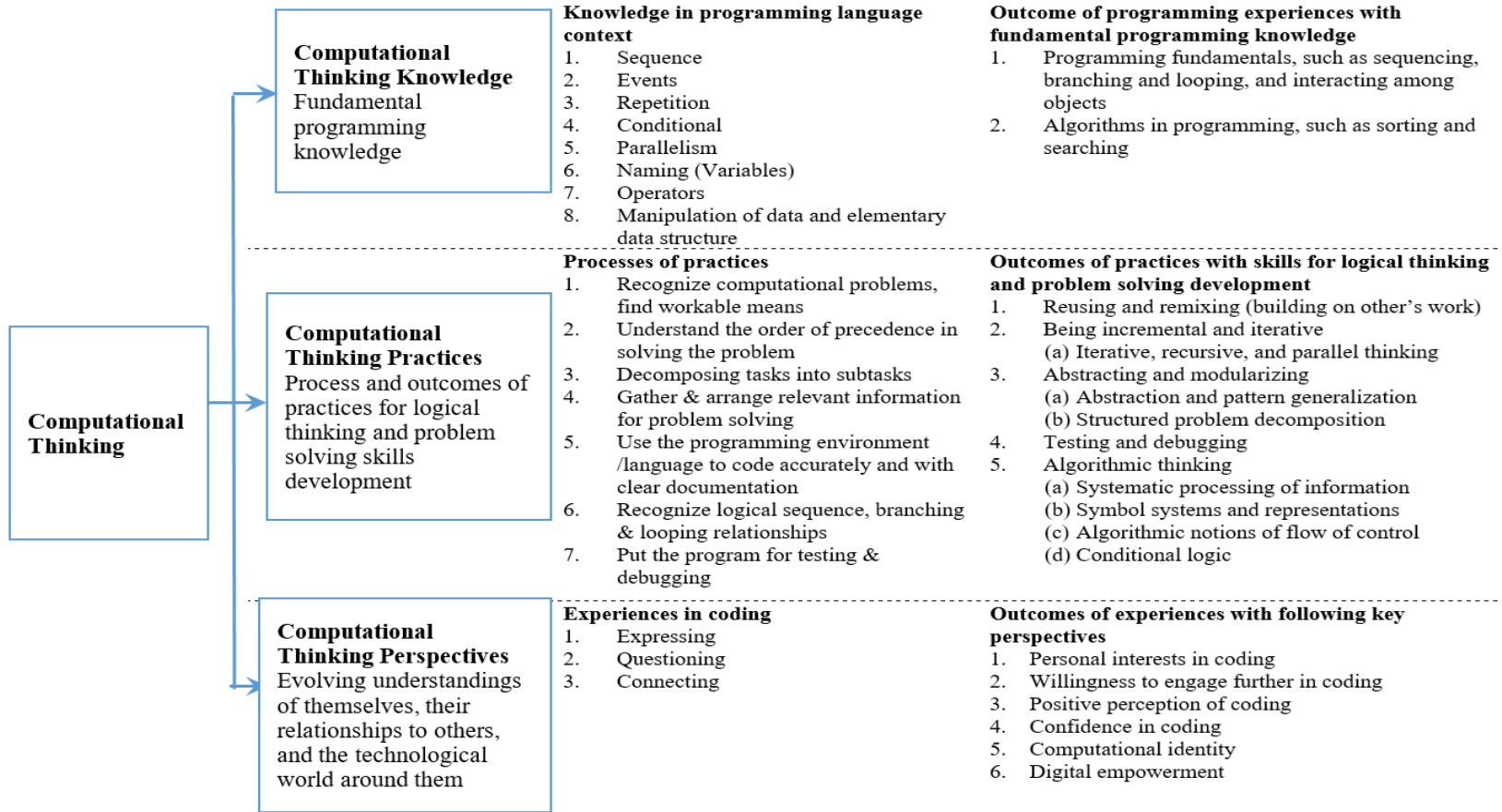
1. Reusing and remixing 重用及整合
2. Being incremental and iterative 反覆及漸進編程
3. Abstracting and modularizing 概念化及模組化
4. Testing and debugging 測試及除錯
5. Algorithmic thinking 算法思維

Computational Thinking Perspectives 運算思維視野

1. Expressing 自我表達
2. Connecting 與生活聯繫
3. Questioning 提問與了解
4. Computational identity 運算身份認同
5. Digital empowerment 數碼充權



CT Framework (Professor KONG Siu Cheung, 2016)



課堂教學設計流程(小學)

概念化及模組化

- 教師展示泥土濕度感應器，並試行運作
- 學生透過工作紙製作流程圖，嘗試把microbit的運作方式模組化

重用及整合

- 教師給予基礎microbit編程碼，學生利用基礎編程碼進行改良

反覆及漸進編程

- 學生把所學的編程碼重整，並有次序地編排事件(Event)發生

算法思維

- 在製作過程中，學生需記錄泥土濕度感應器的結果，透過重整程式碼結構，令濕度感應器能達到預期的表現

測試及除錯

- 學生會先製作原形，並進行測試
- 在測試後會根據所記錄的結果進行改良

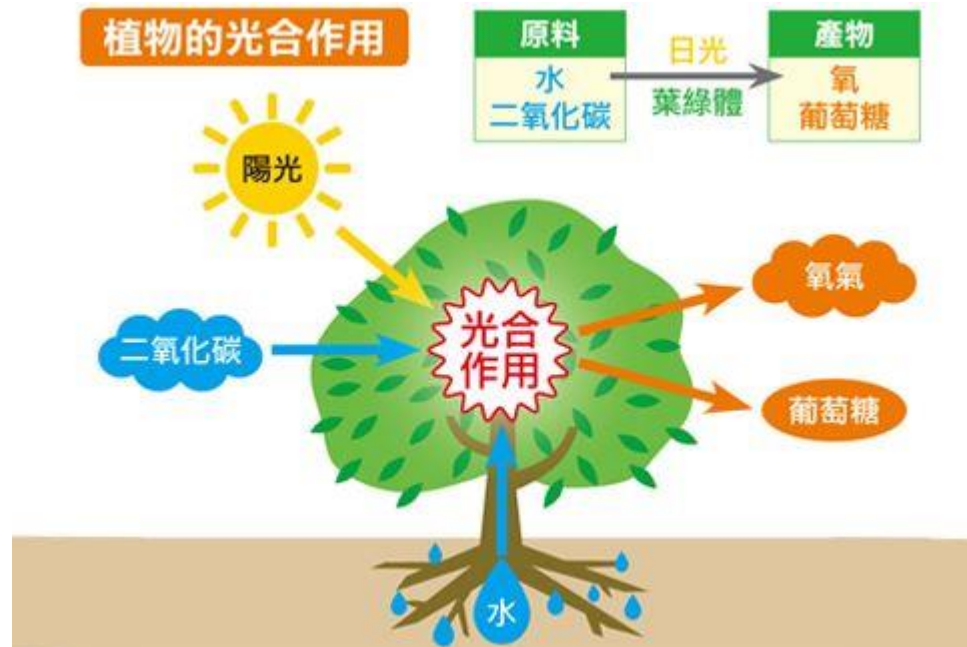


BBC microbit課堂應用實例 (泥土濕度感應器)



課題選擇的考慮因素

1. 是否合適運用編程？
2. 課題是否能提升學生對課題的了解？
3. 能否引起學生的學習興趣？
4. 學生是否有能力去應付？



學生已有知識

- 1.學生已完成植物與環境的課堂
- 2.學生於電腦課已學習**blockly**編程
- 3.學生有運用**microbit**的經驗



教學流程

- 1.教師講解課堂目的，展示泥土濕度感應器，並示範如何量度泥土濕度。
- 2.教師引導學生利用工作紙繪製出程式碼的流程圖。
- 3.學生組裝microbit泥土濕度感應器
- 4.教師給予基礎編程碼
- 5.學生利用基礎編程碼配合流程圖進行編程
- 6.進行原形製作
- 7.測試原形
- 8.改良作品
- 9.進行實地測試
- 10.再改良作品
- 11.匯報成果



為何不同植物的濕度會不同呢？



學生在製作過程中需作的解難

1.泥土濕度感應器所讀的數據問題:

沒水時數值高，有水時數值低。學生需利用數式把數據調至沒水時數值在0-5之間，有水時數值在95-100之間。

2.不同的植物所需的水份不同問題:

學生需進行實際數據測試，並了解該種植物的特性才能改良。



總結

- 學生能掌握自主學習的元素
- 作品還有改良的需要，但重點是他們透過學習綜合應用**stem**元素
- 學生能把課堂的學習模式轉移至其他學習範疇上



Reference

- Barbara, M., & Donna, V. (2005). Learner-Centered Framework for E-learning. *Teachers college record*, 107(8), 1582-1600.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. M. (2011). Habits of programming in scratch. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education, ITiCSE '11*, pp. 168-172.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: a review and discussion. *Computer Science Education*, 13(2), 137–172.
- Yang, T. C., Hwang, G. J., & Yang, S. J. H. (2013). Development of an Adaptive Learning System with Multiple Perspectives based on Students? Learning Styles and Cognitive Styles. *Educational Technology & Society*, 16(4), 185-200.
- 陳桂涓、鄧廣威、何敏華、劉國賢 (2002) : 《利用個別差異建立學習群體，【個別差異發展及研究報告系列 (初期報告) 】》, 香港，教育署課程發展處，研究、評估與設計組。
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. Retrieved from
- <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- **Kong, S. C.**, & Li, P. (2016, November/December). A case study illustrating coding for computational thinking development. In W. Chen, J.-C. Yang, S. Murthy, S. L. Wong & S. Iyer (Eds.), *Proceedings of the 24th International Conference on Computers in Education ICCE 2016* (pp. 579-584). India: Asia-Pacific Society for Computers in Education.

