

# 運算思維融入程式判斷結構課程之學習成效

## Impacts of Employing Computational Thinking on Learning

### Outcomes in a Control Structure Course

洪阡珈<sup>1</sup> 王健華<sup>2</sup>

HONG, CHIEN JIA<sup>1</sup> WANG, CHANG HWA<sup>2</sup>

<sup>1</sup> 國立臺灣師範大學 圖文傳播學系碩士班 研究生

<sup>1</sup>National Taiwan Normal University of Department of Graphic Arts and  
Communications Student

E-mail : [60972005h@gapps.ntnu.edu.tw](mailto:60972005h@gapps.ntnu.edu.tw)

<sup>2</sup> 國立臺灣師範大學 圖文傳播學系碩士班 教授

<sup>2</sup>National Taiwan Normal University of Department of Graphic Arts and  
Communications Professor

E-mail : [wangc@ntnu.edu.tw](mailto:wangc@ntnu.edu.tw)

### 摘要

程式設計已成為全球所重視的教育，因此教學方式與教學內容都是發展的重點項目。本研究旨在探討融入運算思維於程式判斷結構課程，對於學習者的程式設計學習成效影響。採準實驗研究方法，設計實驗組（融入運算思維於課程之教學方式）、對照組（傳統教學方式）的教學安排，以自製「學習成效測驗試題」、「學習成效評分表」作為學習成效的研究工具，並透過統計分析本研究之教學設計對於學習者學習成效的影響。研究結果顯示，本研究之教學設計能提升學習者的學習成效。

**關鍵字：**運算思維、判斷結構、程式設計、學習成效

### Abstract

Programming has become a global focus of education, so teaching methods and content are the focus of development. The purpose of this study was to investigate the impact of incorporating operational thinking into the course of programming judgment structure on the learning of programming. Using the Quasi-experimental research method, the teaching arrangement of the experimental group (placed computational thinking on course) and the control group (normal teaching methods) was designed, and the self-made ‘learning outcomes test’

and “learning outcomes evaluation table” were used as the research tools of learning outcomes, and the influence of the teaching design on learners' learning outcomes was statistically analyzed. The results show that the instructional design of this research can improve the learning outcomes of learners.

**Keywords** : Computational Thinking, Control Structure, Computer Programming, Learning Outcomes

## 壹、前言

### 一、研究背景與動機

資訊科技發達的時代背景下，程式設計課程為教育的一大重點。程式設計是一複雜且具順序性的知識，基本概念知識會影響著後續的學習。如果在學習程式基本概念時，就無法理解其概念，那將會影響後續的學習，形成一個惡行循環（Savage & Piwek, 2019）。將想法轉換為抽象的程式語言是學習者學習程式的困難之處，因此如何「解決問題」是程式學習的挑戰（Medeiros, Ramalho, & Falcão, 2018）。然而，教學教材及教學方法的設計，亦須搭配得宜才能解決程式學習的困境（Cheah, 2020）。

多數國家將程式教育目標訂定為培養學生「解決問題」的能力，「運算思維」的概念也因此成為學習內容之一。運算思維是「解決問題」的思考流程，此過程具有邏輯系統性。而控制結構是程式的基本概念之一，為程式執行流程的架構，程式流程的控制會影響著程式執行結果，如果能將程式流程控制得當，不僅能讓程式正確執行，亦能獲得更符合預期的結果。

綜合上述，運算思維為一有邏輯性的思考流程，程式控制結構為程式執行的流程，兩者皆為有系統及目標性的流程，且目標皆為成功「解決問題」。因此，本研究欲將運算思維融入於程式控制結構的課程中，探討此種教學方法的設計，對於學習程式設計之學習成效的影響，希冀研究結果能提供程式教育另一種教學方法的參考。

### 二、研究目的

- (一) 運算思維融入程式判斷結構之教學方式，對於學習者學習成效影響。
- (二) 運算思維融入程式判斷結構之教學方式，對於 if...或 if...else 程式結構正確性之影響。
- (三) 運算思維融入程式判斷結構之教學方式，對於符號正確性之影響。
- (四) 運算思維融入程式判斷結構之教學方式，對於縮排格式正確性之影響。
- (五) 運算思維融入程式判斷結構之教學方式，對於條件式正確性之影響。

### 三、研究問題

根據研究目的，形成以下研究問題：

1. 運算思維融入程式判斷結構之教學方式，對於學習者學習成效之影響為何？
2. 運算思維融入程式判斷結構之教學方式，對於 if...或 if...else 程式結構正確性之影響為何？
3. 運算思維融入程式判斷結構之教學方式，對於符號正確性之影響為何？
4. 運算思維融入程式判斷結構之教學方式，對於縮排格式正確性之影響為何？
5. 運算思維融入程式判斷結構之教學方式，對於條件式正確性之影響為何？

## 貳、文獻探討

### 一、運算思維

#### (一) 運算思維定義與核心元素

運算思維一詞之定義由周以真教授於 2006 年提出。Wing (2017) 將運算思維之定義具體化，認為運算思維是以電腦（人或機器）可有效執行的角度，去理解問題並思考解決方案的思維過程。學者們開始探討運算思維的組成要素，美國國際社會教育科技組織 (ISTE) 和電腦科學教師協會 (CSTA) 提出運算思維的操作性定義，將運算思維分為創造力、演算法思考、批判性思考、問題解決、合作等五元素 (ISTE & CSTA, 2011)。運算思維流程有四個核心元素，為根據程式設計流程中的含義所定義出的核心元素。四核心元素分別為問題拆解、規律辨識、摘要化、步驟化。以下為 British Broadcasting Corporation (BBC) 對運算思維四元素的定義：

1. 問題拆解 (Decomposition)：  
將問題細分為一系列的子問題，或是統整相似的部分，形成一個易於執行的問題。
2. 規律辨識 (Pattern Recognition)：從細分出來的子問題中，找出相似之處，並思考這些相似問題的解決方式是否相同。
3. 摘要化 (Abstraction)：統整歸納出問題解決方法的核心之處為何。
4. 步驟化 (Algorithm)：設計一個解決問題的方法，並依據方法步驟逐一執行，來解決問題。

### 二、程式控制結構

#### (一) 控制結構定義與特性

具有控制結構的程式稱之為「結構化程式設計 (Structured Programming)」。結構化的程式設計為一種程式設計的方法，它能提升程式的可讀性、執行時間 (Corrado Böhm & Giuseppe Jacopini, 1966)，因此控制結構可視為模組，程式會依據這個模組來執行，也就是程式執行的流程。

程式控制結構有循序結構 (Sequence Structure)、判斷結構 (Decision Structure)、重複結構 (Loop) 三種。根據 Python 的語言參考手冊對三種控制結構的定義為：

(1) 循序結構 (Sequence Structure)：程式依照程式語法的順序，依序執行。

(2) 判斷結構 (Decision Structure)：程式執行時，會依據使用者輸入的資料進行判斷，執行對應的程式區塊。因此在判斷結構中，需有條件式的設計

(3) 重複結構 (Loop)：當設定的條件成立時，會重複執行該區塊的程式。

綜合上述，透過運算思維來解決問題的過程是以電腦科學的角度去思考，而判斷結構為程式控制結構的一環，為控制程式執行流程的結構。程式的抽象概念理解是學習者認為學習程式的困難之一，因此本研究欲融入運算思維於程式判斷結構課程，給予學習者有邏輯性的教學內容，來探討其學習成效。

## 參、研究實施與設計

### 一、研究方法

本研究採取準實驗的設計，實作組為融入運算思維於課程，對照組則無。選擇程式控制結構中的判斷結構作為教學主題，並選擇判斷結構中的 if...與 if...else...兩種程式敘述進行教學內容的設計。透過前、後測的設計，探討將運算思維融入判斷結構程式課程中，對於學習者的學習成效之影響。

### 二、研究對象

本研究的研究對象為普通型高中一年級學生，兩個班級總人數為 66 人。採非隨機分組的方式，實驗組 34 人，對照組 32 人。全體學習者的學習時間、內容皆相同。

### 三、研究工具

本研究之研究工具為有「Python 語言(3.8.8)」、「Python IDE - Spyder(4.2.5)」、「學習成效前、後測試卷」、「學習成效測驗試題評分表」，以下針對學習成效前、後測試題、學習成效測驗試題評分表分列詳述。

#### 一、學習成效前、後測試題

學習成效將會分析學生在課程前、後測之成績表現。研究者針對課程內容進行試卷的出題，並邀請三位專家為試題進行審查與修訂，來建立本試卷專家效度。本試卷共五題實作題，每題四分，總分為二十分，詳細內容於附錄一。

#### 二、學習成效測驗試題評分表

學習成效測驗試題評分表為學習者在測驗試題作答內容的評分標準。根據學生作答狀況，與專家們討論後統整出四項錯誤項目，並將此四項目列為評分項目。此四項評分項目為：if...或 if...else 程式結構錯誤、符號錯誤、縮排格

式錯誤、條件式錯誤。專家們將依據評分表項目進行評分，每一個項目皆為一分，只要有一錯誤，該項目即得零分，因此每一評分項目的總分為五分，詳細的評分表內容如表 3.1。

表 3.1 學習成效測驗評分表

判斷結構 (4分)	
評分項目	得分
1.1 if...或 if...else 程式結構錯誤	
1.2 符號錯誤	
1.3 縮排格式錯誤	
1.4 條件式錯誤	

#### 四、教學設計

依據課程主題，研究者設計兩大教學活動。一為程式判斷結構的概念講解；二為程式判斷結構的實作練習。以下將分別說明分別教學活動的內容：

##### 一、程式判結構概念

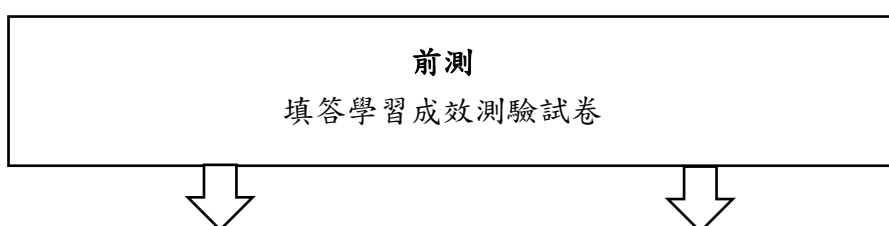
此教學活動設計在於講解判斷結構的概念，以及透過日常生活中的例子來讓學習者理解判斷結構在日常生活中的應用為何。

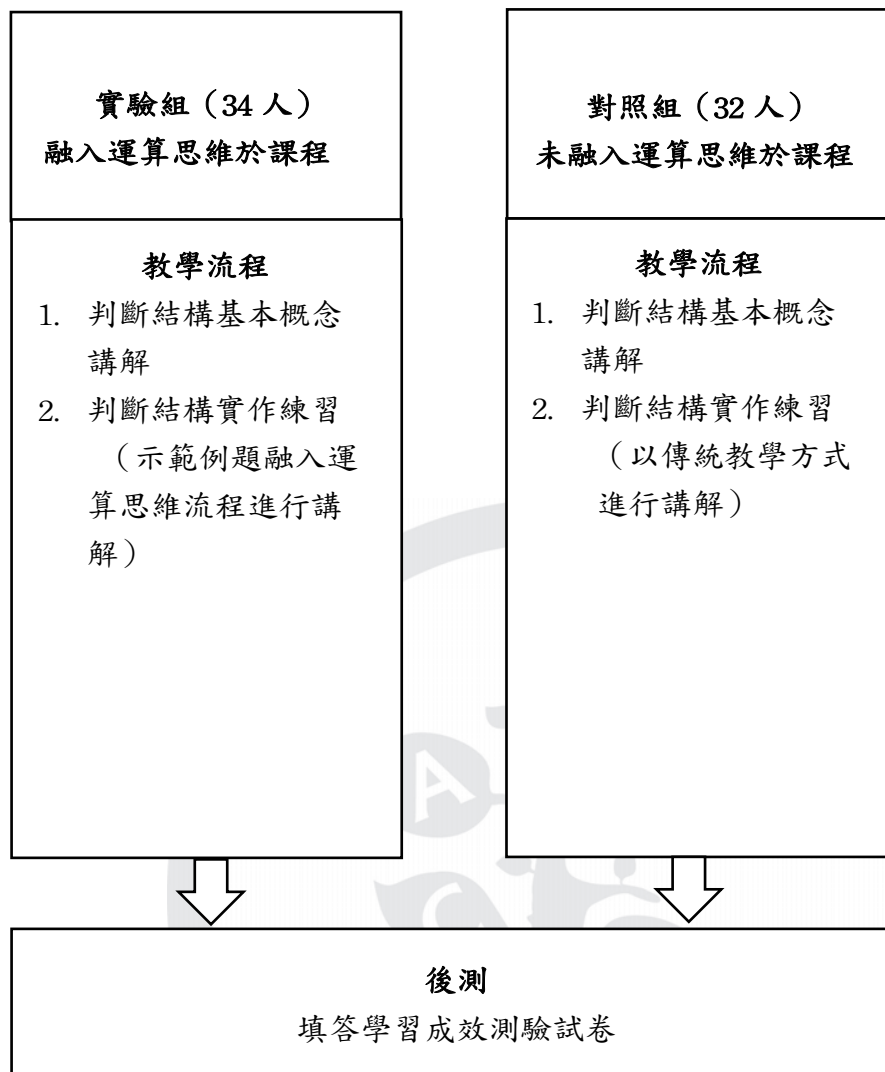
##### 二、程式判斷結構實作練習

基本概念講解完後，此教學活動將讓學習者透過上機實作來編寫程式碼，將想法轉換為程式碼，並將結果輸出。在學習者實作練習前，教學者會先實作示範例題，在示範例題的過程中，會以運算思維流程所包含的四步驟來進行示範講解。練習題共計三題。以下將說明運算思維在本課程中所代表的含義：

- 1.問題拆解：確定程式的判斷目標（期望達到的結果為何），再根據判斷的目的來列出對應的條件式有哪些。
- 2.規律辨識：從條件式中來設計條件式的成立與否，各自對應的結果有哪些。
- 3.摘要化：依照設計的條件式各自對應的結果，來組成一個判斷結構。
- 4.步驟化：依照程式執行流程，將判斷條件、欲達到的結果依序代入其中，來設計程式。

#### 五、研究實施





## 肆、研究結果與討論

本研究探討將運算思維融入程式判斷結構課程，對於學習者學習成效之影響，並依據程式判斷結構之程式敘述，細分出四個評分項目，分別為 if...與 if...else 程式判斷結構、符號、縮排格式、條件式。因此學習成效除了依據測驗試題的總分作為統計數據，也將分別根據學習者在四個評分項目的得分進行統計分析與探討。本研究之對照組共 32 人，實驗組共 34 人，經統計後共有 66 筆有效數據，將以 IBM SPSS Statistics 26 統計軟體進行數據統計分析。以下將根據統計分析結果來逐一回答研究問題。

### 一、有無融入運算思維於程式判斷結構課程之學習成效分析

此小節為了解學習者在進行融入運算思維的程式判斷結構課程後，是否有提升學習成效。實驗組、對照組兩組別在學習成效前測分數皆為 0 分，因此根據兩組別在學習成效後測分數，進行獨立樣本 T 檢定，比較兩組的差異，來回

答研究問題一：「運算思維融入程式判斷結構之教學方式，對於學習者學習成效之影響為何？」下表 4-1 為實驗組、對照組在學習成效前、後測之敘述性統計分析表，表 4-2 為實驗組、對照組學習成效後測之獨立樣本 T 檢定。

表 4-1 實驗組、對照組學習成效前、後測之敘述性統計分析表

		N	Mean	SD
前測	實驗組	34	0.00	0.00
	對照組	32	0.00	0.00
後測	實驗組	34	12.18	1.524
	對照組	32	10.34	2.664

表 4-2 實驗組、對照組學習成效後測成績之獨立樣本 T 檢定分析表

	t	df	p
不同組別學習成效後測成績	4.742	53.110	.000

由表 4-1 敘述性統計結果可看出實驗組、對照組在學習成效後測的平均數，實驗組高於對照組。另以表 4-2 的統計結果可看出，兩個組別之間的學習成效後測成績有顯著差異，即代表有無融入運算思維於程式判斷結構課程，對於學習成效有顯著影響。

## 二、if...或 if...else 程式結構正確性之學習成效分析

此小節為了解學習者在進行融入運算思維的程式判斷結構課程後，是否有提升 if...或 if...else 程式結構正確性。實驗組、對照組兩組別在學習成效前測分數皆為 0 分，因此根據兩組別在 if...或 if...else 程式結構評分項目之後測分數，進行獨立樣本 T 檢定，比較兩組的差異，來回答研究問題二：「運算思維融入程式判斷結構之教學方式，對於 if...或 if...else 程式結構正確性之影響為何？」下表 4-3 為實驗組、對照組在 if...或 if...else 程式結構後測成績之敘述性統計分析表，表 4-4 為實驗組、對照組在 if...或 if...else 程式結構後測成績之獨立樣本 T 檢定。

表 4-3 不同組別在 if...或 if...else 程式結構後測成績之敘述性統計分析表

		N	平均值	標準差
if...或 if...else 程式結構後測成績	實驗組	34	4.9118	.28790
	對照組	32	3.7500	1.16398

表 4-4 不同組別在 if...或 if...else 程式結構後測成績之獨立樣本 T 檢定分析表

	t	df	p
不同組別「if...與 if...else...判斷結構程式」後測成績	5.490	34.565	.000

由表 4-3 敘述性統計分析結果看出實驗組後測成績平均高於對照組，再由表 4-4 的統計結果可看出，兩組別在「if...與 if...else...判斷結構程式」評分項目的後測成績有顯著差異，即代表有無融入運算思維於程式判斷結構課程，對於「if...與 if...else...判斷結構程式」有顯著影響。

### 三、符號正確性之學習成效分析

此小節為了解學習者在進行融入運算思維的程式判斷結構課程後，是否有提升符號正確性。實驗組、對照組兩組別在學習成效前測分數皆為 0 分，因此根據兩組別在符號評分項目之後測分數，進行獨立樣本 T 檢定，比較兩組的差異，來回答研究問題三：「運算思維融入程式判斷結構之教學方式，對於符號正確性之影響為何？」下表 4-5 為實驗組、對照組在符號後測成績之敘述性統計分析表，表 4-6 為實驗組、對照組在符號評分項目後測成績之獨立樣本 T 檢定分析表。

表 4-5 不同組別在符號後測成績之敘述性統計分析表

	組別	N	平均值	標準差
符號後測成績	實驗組	34	3.3235	1.34221
	對照組	32	2.6563	1.28539

表 4-6 不同組別在符號後測成績之獨立樣本 T 檢定分析表

	t	df	p
不同組別「符號」後測成績	2.060	64	.043

由表 4-5 敘述性統計分析結果看出實驗組後測成績平均高於對照組，再由表 4-6 的統計結果可看出，兩組別在「符號」評分項目的後測成績有顯著差異，即代表有無融入運算思維於程式判斷結構課程，對於「符號」有顯著影響。

### 四、縮排格式正確性之學習成效分析

此小節為了解學習者在進行融入運算思維的程式判斷結構課程後，是否有提升縮排格式正確性。實驗組、對照組兩組別在學習成效前測分數皆為 0 分，因此根據兩組別在縮排格式評分項目之後測分數，進行獨立樣本 T 檢定，比較兩組的差異，來回答研究問題四：「運算思維融入程式判斷結構之教學方式，對於縮排格式正確性之影響為何？」下表 4-7 為實驗組、對照組在縮排格式後測成績之敘述性統計分析表，表 4-8 為實驗組、對照組在縮排格式評分項目後測成績之獨立樣本 T 檢定分析表。

表 4-7 不同組別在縮排格式後測成績之敘述性統計分析表

	組別	N	平均值	標準差
縮排格式後測 成績	實驗組	34	2.7941	1.40948
	對照組	32	2.3750	1.09985

表 4-8 不同組別在縮排格式評分項目後測成績之獨立樣本 T 檢定分析表

	t	df	p
不同組別「縮排 格式」後測成績	1.341	64	.185

由表 4-7 敘述性統計分析結果看出實驗組後測成績平均高於對照組，進一步分析由表 4-8 的統計結果可看出，兩個組別之間在「縮排格式」評分項目的後測成績無顯著差異，表示縮排格式正確性可能並非受到有無融入運算思維於程式判斷結構課程所影響，須進一步探討其他原因。

## 五、條件式正確性之學習成效分析

此小節為了解學習者在進行融入運算思維的程式判斷結構課程後，是否有提升條件式正確性。實驗組、對照組兩組別在學習成效前測分數皆為 0 分，因此根據兩組別在條件式評分項目之後測分數，進行獨立樣本 T 檢定，比較兩組的差異，來回答研究問題五：「運算思維融入程式判斷結構之教學方式，對於條件式正確性之影響為何？」下表 4-9 為實驗組、對照組在條件式後測成績之敘述性統計分析表，表 4-10 為實驗組、對照組在條件式評分項目後測成績之獨立樣本 T 檢定分析表。

表 4-9 不同組別在條件式後測成績之敘述性統計分析表

	組別	N	平均值	標準差
條件式後測成 績	實驗組	34	1.7353	1.33278
	對照組	32	1.4688	1.01550

表 4-10 不同組別在條件式評分項目後測成績之獨立樣本 T 檢定分析表

	t	df	p
不同組別「條件式」後測成績	.910	64	.366

由表 4-9 敘述性統計分析結果看出實驗組後測成績平均高於對照組，進一步分析由表 4-10 的統計結果可看出，兩個組別之間在「條件式」評分項目的後測成績無顯著差異，表示條件式正確性可能並非受到有無融入運算思維於程式判斷結構課程所影響，須進一步探討其他原因。

## 伍、結論與建議

### 一、研究結論

以下將依據研究問題，來依序回答出本研究之結論。

(一) 運算思維融入程式判斷結構之教學方式，對於學習者學習成效之影響為何？

實驗組、對照組在進行教學課程前的前測分數均為 0 分，而在進行教學課程後的後測分數有顯著差異。由此可得出將運算思維融入於程式的判斷結構課程中，會顯著影響學習者之學習成效。

(二) 運算思維融入程式判斷結構之教學方式，對於 if... 或 if...else 程式結構正確性之影響為何？

實驗組、對照組在進行教學課程前的「if... 或 if...else 程式結構」評分項目之前測分數均為 0 分，而在進行教學課程後的後測分數有顯著差異。由此可得出將運算思維融入於程式的判斷結構課程中，會顯著影響學習者在 if... 或 if...else 程式結構之正確性。

(三) 運算思維融入程式判斷結構之教學方式，對於符號正確性之影響為何？

實驗組、對照組在進行教學課程前的「符號」評分項目之前測分數均為 0 分，而在進行教學課程後的後測分數有顯著差異。由此可得出將運算思維融入於程式的判斷結構課程中，會顯著影響學習者在符號之正確性。

(四) 運算思維融入程式判斷結構之教學方式，對於縮排格式正確性之影響為何？

實驗組、對照組在進行教學課程前的「縮排格式」評分項目之前測分數均為 0 分，而在進行教學課程後的後測分數並無顯著差異。在縮排格式的評分項目中，學習者須在判斷結構之程式敘述透過縮排來區分出程式的執行區塊。根

據全體學習者的作答內容，發現學習者在縮排格式的錯誤之處在於 if...判斷結構外的縮排、if...else...結構內的縮排。

因此對於融入運算思維於程式判斷結構課程，針對縮排格式之教學的設計，有待進一步探討。

(五) 運算思維融入程式判斷結構之教學方式，對於條件式正確性之影響為何？

實驗組、對照組在進行教學課程前的「條件式」評分項目之前測分數均為 0 分，而在進行教學課程後的後測分數並無顯著差異。

在條件式的評分項目中，學習者須根據題目的要求設計對應的條件式。根據全體學習者的作答內容，發現學習者在條件式的設計，錯誤之處在於運算子的使用，條件式之程式敘述會用到程式的概念有「邏輯運算子、算數運算子、關係運算子」。

因此，推測在判斷結構的條件式中，先備知識的理解程度會影響學習者設計條件式，此論點需再進一步探討。

## 二、後續研究建議

### (一) 教學課程時間

本研究的教學實驗設計為一節課，檢驗學習者在這一節課的學習成效為何。根據研究結果看出影響學習者在判斷結構的程式敘述之細節（縮排格式、條件式）的表達仍有待進一步探討，因此未來研究可拉長教學時間，藉以觀察學習者之學習成效。

### (二) 教學內容加入先備知識概念

程式設計為複雜、專業性高的知識。根據本研究結果看出學習者在判斷結構的條件式設計時，對於邏輯運算子、算數運算子、關係運算子的表達錯誤，而運算子屬於課程的先備知識內容。因此未來研究可將先備知識納入課程的複習單元，亦可將先備知識作為研究變項做更進一步的探討。

## 參考文獻

Böhm, C., Jacopini, G. Flow Diagrams, Turing Machines and Languages with only Two Formation Rules. *Commun. the ACM*, 9(5), 366-371.

CSTA. (2011). *K-12 computer science standards*. Retrieved December 11, 2016, from <http://csta.acm.org/Curriculum/sub/K12Standards.html>

Cheah, C. S. (2020). Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. *Contemporary Educational Technology*, 12(2), ep272.

ISTE (2011). *Operational definition of computational thinking for K-12 education*. Retrieved from <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>

R. P. Medeiros, G. L. Ramalho, & T. P. Falcão. (2018). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, 62(2), 77-90.

Savage, S., Piwek, P. (2019). Full report on challenges with learning to program and problem solve: an analysis of first year undergraduate Open University distance learning students' online discussions.

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

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7-14.