

Python

Computer Programming and Developing Applications for Education

Python



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

Course: Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

อาจารย์บรรยาย



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

การศึกษา

- 2565 ปริญญาเอก ปรัชญาดุสิตบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตร์มหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตร์บัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขารุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCN-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตร์ข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชิแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

Lecturer



Asst.Prof.Dr.Nutthapat Kaewrattanapat

Education

- 2022 Doctoral Degree: Ph.D. in Information and Communication Technology for Education (GPA. 4.00)
- 2008 Master Degree: M.Sc. in Management Information System (GPA. 3.58)
- 2006 Bachelor Degree: B.Sc. With 1st Class Honors in Computer Science (GPA. 3.66)

Certificates

- Certificate of Professional Qualification Level 6 Sub-Sector: Digital Business and Electronic Commerce Occupation: Business Information Technology Manager, TPQI
- Python for Data Science, **Thammasat University**
- Computer Programming for Linguist, **Mahidol University**
- Introduction to Python Programming, **University of Pennsylvania, U.S.**
- Programming for Everybody, **University of Michigan, U.S.**

Contact: nutthapat.ke@ssru.ac.th

Course Description

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลักการ พัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการ เบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การ วิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมินซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and development applications, computer programming principles, computer language, elements of computer language, syntax, computer programming, algorithms, analysis and design application for education, software evaluation, candidate teachers able to developing applications for education.

Algorithm Design

Python Programming Language

System Development Life Cycle: SDLC

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

Course Outline

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- Chapter 3 – Conditional Statement
- Chapter 4 – Iteration Statement
- Chapter 5 – Strings
- Chapter 6 – Lists, Tuples, Sets, Dictionaries
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP
- Chapter 9 – Python and Database Management System
- Chapter 10 – System Development Life Cycle: SDLC

Measurement and Evaluation

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%
24%
12%
4%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
1 พฤษ 7 รค 66	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ บทที่ 1 การโปรแกรมคอมพิวเตอร์	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
2 พฤษ 14 รค 66	บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
3 พฤษ 21 รค 66	บทที่ 3 การโปรแกรมแบบตัดสินใจ (Decision)	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
4 พฤษ 28 รค 66	บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration)	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
5 พฤษ 4 มค 67	บทที่ 5 การโปรแกรมสายอักขระ (String)	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
6 พฤษ 11 มค 67	บทที่ 6 ลิสต์ ทัวเพิล เซ็ต และดิกชันนารี	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
7 พฤษ 18 มค 67	บทที่ 7 ฟังก์ชัน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
8 พฤษ 25 มค 67	สอบกลางภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			

Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
9 พฤษ 1 กพ 67	บทที่ 8 การโปรแกรมเชิงวัตถุ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
10 พฤษ 8 กพ 67	บทที่ 8 การโปรแกรมเชิงวัตถุ ต่อ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
12 พฤษ 15 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
13 พฤษ 22 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล ต่อ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
14 พฤษ 29 กพ 67	บทที่ 10 วงจรการพัฒนาระบบ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
15 พฤษ 7 มีค 67	สอบปลายภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
16 พฤษ 14 มีค 67	ส่งโครงงาน และนำเสนอ 20%			
17 พฤษ 21 มีค 67				

Pretest

1. ภาษา Python เป็นภาษาคอมพิวเตอร์ลักษณะใด?

A

Interpreter

B

Compiler

2. ข้อใดไม่เป็นองค์ประกอบของระบบคอมพิวเตอร์

A

Hardware

D

Software

B

Peopleware

E

Data/Information

C

Communication

F

Procedures

3. "110100101010101" เทียบเคียงได้กับภาษาคอมพิวเตอร์ระดับใด?

A

Low-Level Language

B

High-Level Language

C

Assembly Language

D

Procedural Language

4. `print("Panda")` เทียบเคียงได้กับภาษาคอมพิวเตอร์ระดับใด?

A

Low-Level Language

B

High-Level Language

C

Assembly Language

D

Procedural Language

5. ข้อใด ไม่ใช่ คุณสมบัติของอัลกอริทึม

A

ไม่คลุมเครือ (Unambiguous)

B

มีขั้นตอนตายตัว (Deterministic)

C

มีจุดสิ้นสุดการทำงาน (Finiteness)

D

มีหน่วยความจำที่เพียงพอ (Memory Capacity)

6. ข้อใดไม่ใช่เครื่องมือในการโปรแกรมภาษา Python

A

Pycharm

B

Colab

C

Jupyter

D

Assembler

7. ให้ผลลัพธ์ถูกต้องตามเงื่อนไขที่กำหนดไว้เสมอ คือ คุณสมบัติข้อใด

A

ไม่คลุมเครือ (Unambiguous)

B

มีขั้นตอนตายตัว (Deterministic)

C

มีจุดสิ้นสุดการทำงาน (Finiteness)

D

มีหน่วยความจำที่เพียงพอ (Memory Capacity)

8. วิธีการสอนใดเหมาะกับการเรียนรู้ด้านการโปรแกรมคอมพิวเตอร์

A

Passivity-based Learning

B

Cognitivism

C

Experiential Learning

D

Adaptive Learning

9. การสอน Coding สิ่งแรกที่ต้องสอนให้ผู้เรียนคำนึงถึงในการโปรแกรม คืออะไร

A

Input

B

Process

C

Output

D

Memory

10. การสอน Coding สิ่งสุดท้ายที่ต้องสอนให้ผู้เรียนคำนึงถึงในการโปรแกรมคืออะไร

A

Input

B

Process

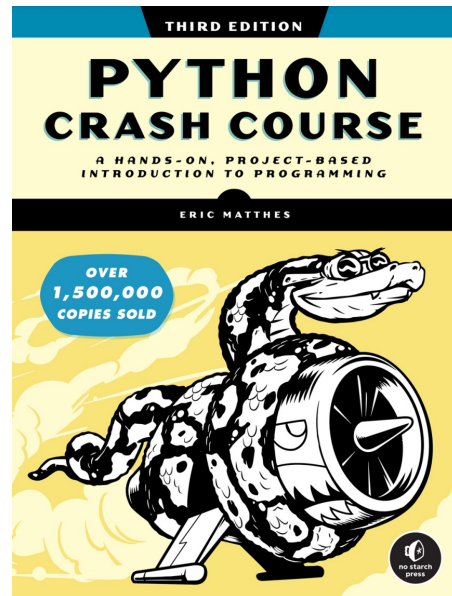
C

Output

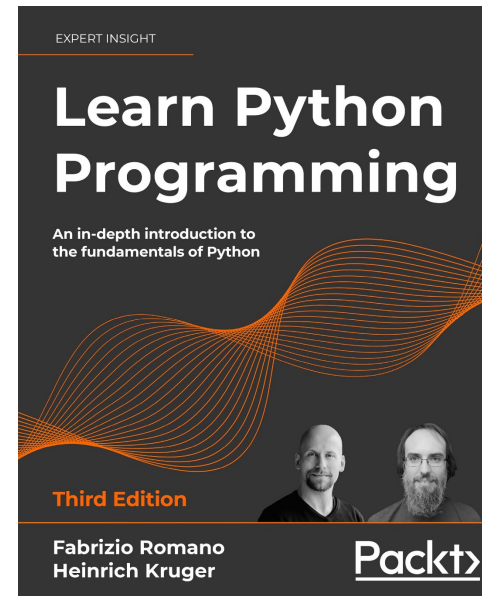
D

Memory

Learning Materials



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Learning Data Science with Python - Tools

Python
Fundamental



<https://colab.research.google.com/>

Python &
Application



Database
Management
System



Learning by Doing – Experiential Learning

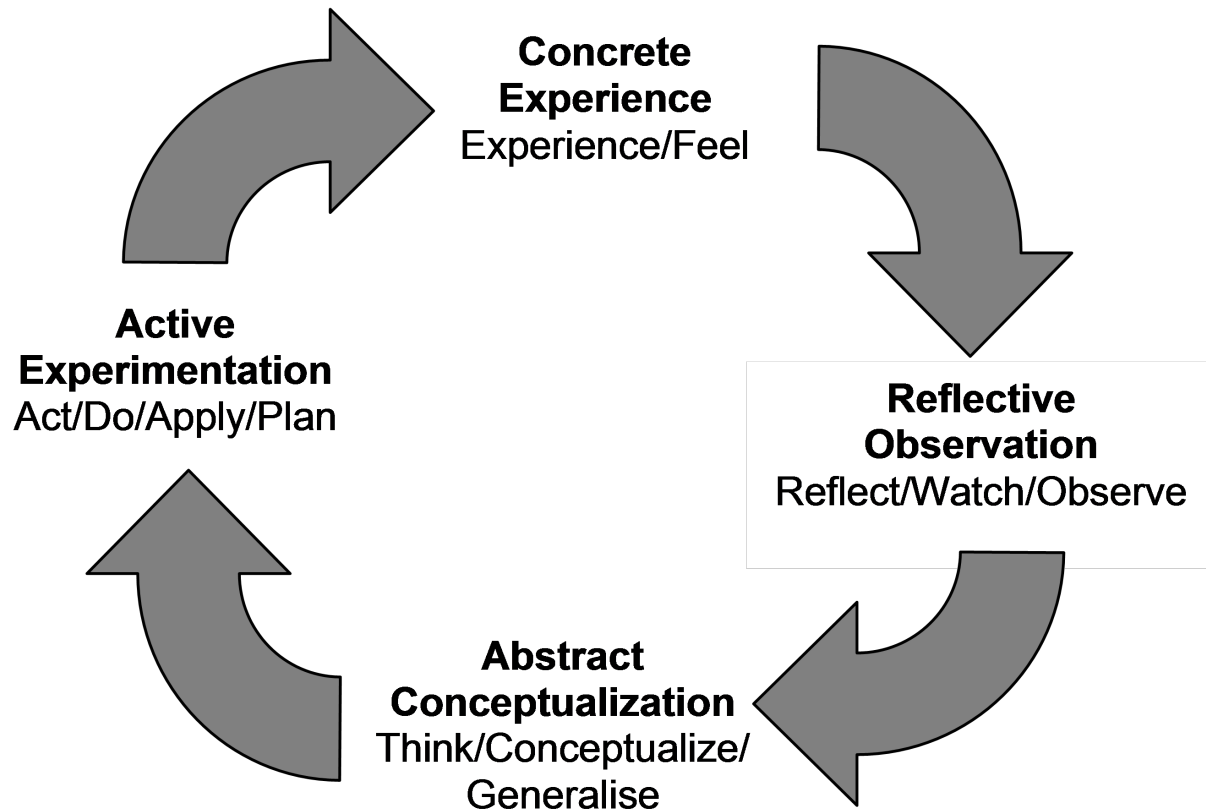
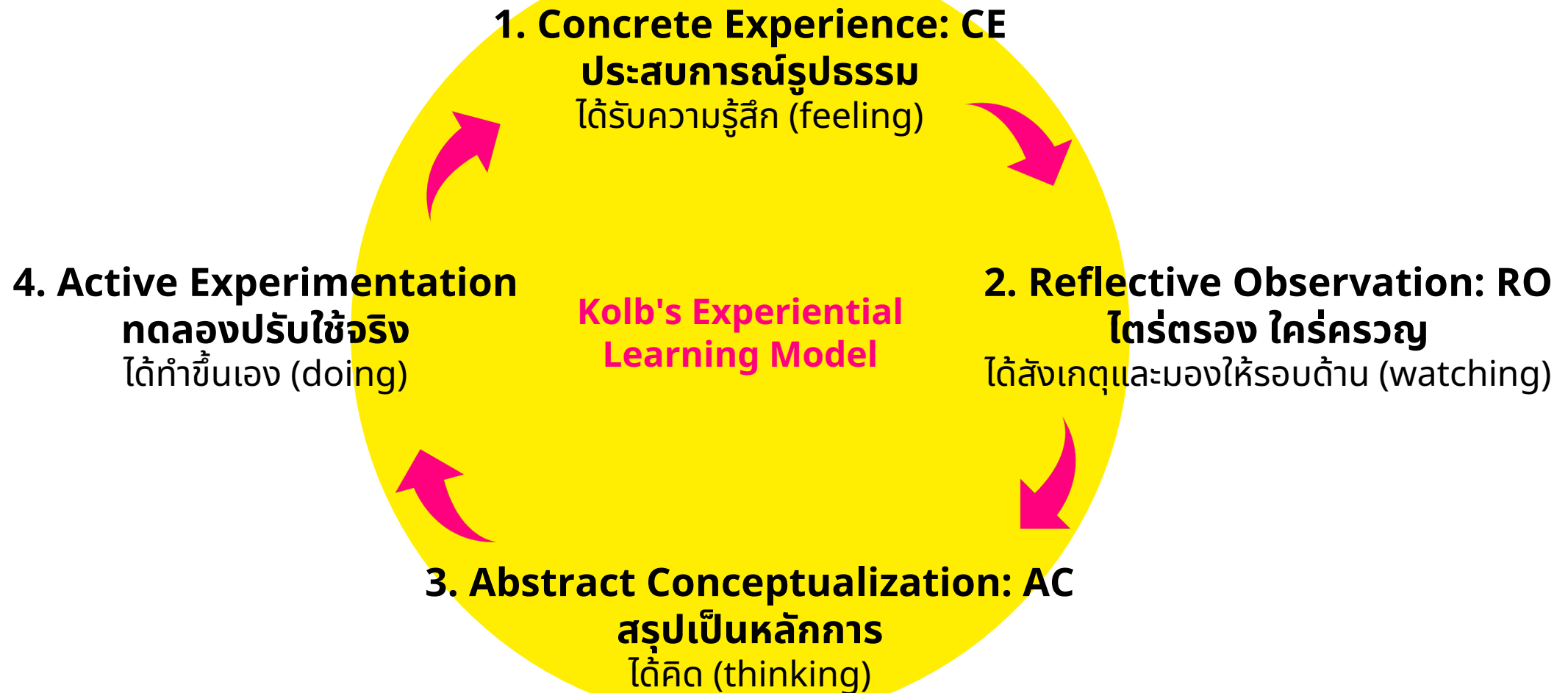


Figure 1. Model of Kolb's (1984) Experiential Learning

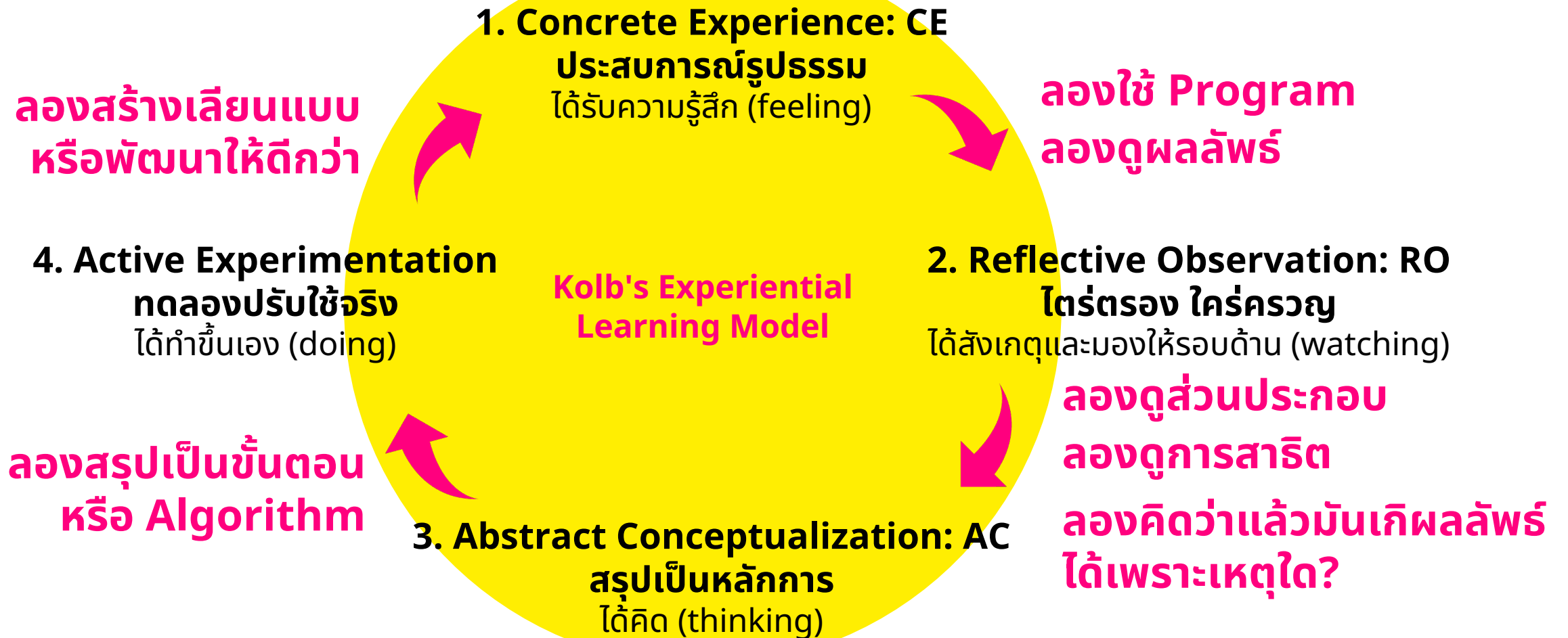
Experiential learning หมายถึงการเรียนรู้ที่เกิดขึ้นจากประสบการณ์ที่ได้ทำและปฏิบัติจริง ไม่ใช่แค่การเรียนรู้จากหนังสือหรือบทเรียนที่ถูกสอนเท่านั้น เมื่อผู้เรียนได้มีโอกาสสัมผัสและทดลองทำสิ่งต่างแล้ว จะช่วยให้เกิดการเรียนรู้ที่มีความหมายและจดจำได้มากขึ้น เช่น

- การศึกษาด้วยการทำโครงการ (Project)
- การฝึกทักษะทางวิชาชีพ หรือ
- การเรียนรู้ผ่านการทำกิจกรรมที่มีจุดมุ่งหมายในการสร้างประสบการณ์และความรู้ใหม่ ๆ ผ่านการทดลองและปฏิบัติจริงในชีวิตประจำวันของตนเอง

Learning by Doing – Experiential Learning



Experiential Learning & Computer Programming



Experiential Learning & Computer Programming

ขั้นตอนที่
1

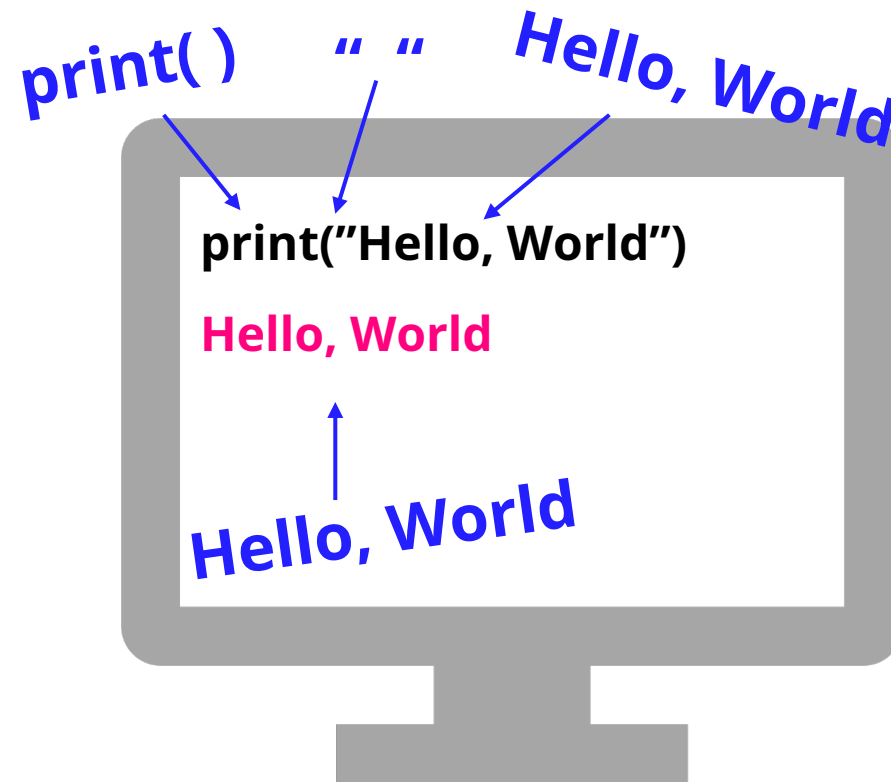
Concrete Experience: CE
ประสบการณ์รูปธรรม



Experiential Learning & Computer Programming

ขั้นตอนที่
2

Reflective Observation: RO
ไตร่ตรอง ใคร่ครวญ



Experiential Learning & Computer Programming

ขั้นตอนที่
3

Abstract Conceptualization: AC
สรุปเป็นหลักการ



มันคือ Syntax

print("ข้อความ")

ข้อความ

Experiential Learning & Computer Programming

ขั้นตอนที่
4

Active Experimentation
ทดลองปรับใช้จริง

```
print("Hello, Nutthapat")  
print("How are you?")
```

Hello, Nutthapat
How are you?

Emoji นศ.ครู วิชาวิทยาการคำนวณ



**Course
Syllabus**



**Hello
World!**



Algorithm



Bug!



Project



**The Project
is finished**

Computer Programming Chapter 1



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

Course: Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

Chapter 1 – Computer Programming

Computer programming refers to the process of designing and creating sequences of instructions (code) that enable a computer to perform specific tasks or functions.

It involves writing and organizing instructions that a computer can understand and execute to achieve desired outcomes.

Programs are typically written in **programming languages such as Python, Java, C++, and others**. These languages use specific syntax and rules to communicate with computers, enabling developers to create applications, websites, software, or systems ranging from simple to complex.

การเขียนโปรแกรมคอมพิวเตอร์ หมายถึง การออกแบบและสร้างลำดับขั้นตอนของคำสั่ง (อัลกอริทึม และโค้ด) ที่ทำให้คอมพิวเตอร์สามารถทำงานหรือปฏิบัติการกิจเฉพาะตามที่กำหนดได้

การเขียนโปรแกรมคอมพิวเตอร์ เป็นกระบวนการที่เกี่ยวข้องกับการเขียนและจัดระเบียบคำสั่งที่คอมพิวเตอร์สามารถเข้าใจและดำเนินการตามเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ

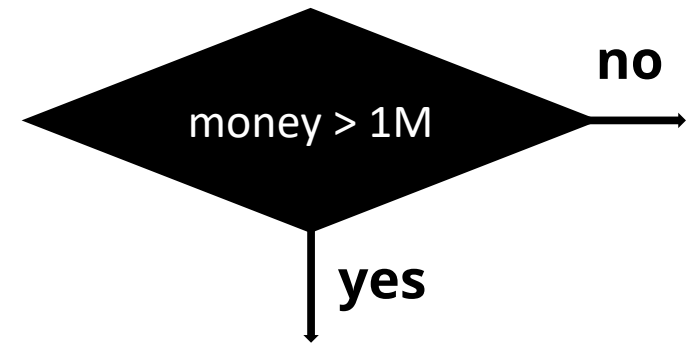
โปรแกรมทั่วไปจะถูกเขียนขึ้นด้วยภาษาโปรแกรม เช่น Python, Java, C++ และอื่น ๆ ซึ่งภาษาเหล่านี้ใช้ไวยากรณ์และกฎเฉพาะเพื่อสื่อสารกับคอมพิวเตอร์ ทำให้นักพัฒนาสามารถสร้างแอปพลิเคชัน เว็บไซต์ ซอฟต์แวร์ หรือระบบต่าง ๆ ที่มีความซับซ้อนตั้งแต่ง่ายไปจนถึงยากได้

โค้ดถึง Coding คือข้อใด?

A

10011100110101

C



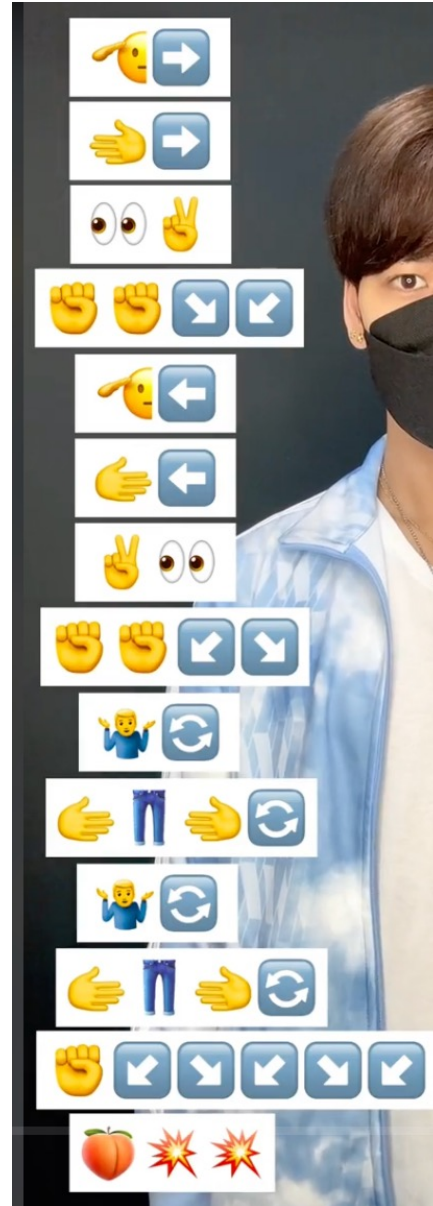
B

print("Hello World!")

D

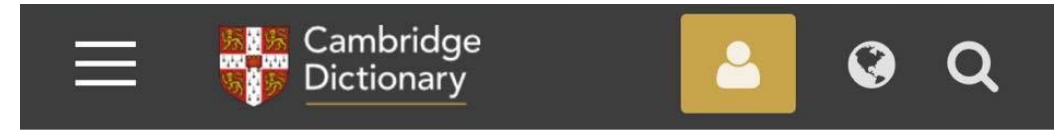


ทำเต็มเพลงอะไร?



โค้ดดิ้ง (Coding)

- Code คือ ระบบของคำ อักษร หรือสัญลักษณ์ ในการแทนข้อความในรูปแบบหนึ่งๆ หรืออาจเป็นระบบจำนวน/ตัวเลข อักษร หรือสัญลักษณ์ที่ใช้แทนอะไรบางอย่างในลักษณะที่สั้นกระชับ หรือ อยู่ในรูปแบบที่เข้าใจได้สะดวกขึ้น
- ดังนั้น พวก รหัส, เลขฐาน, สูตร, Pseudocode, Flowchart, Diagram, Computer Programming ต่าง ๆ นับว่าเป็น Code ในลักษณะหนึ่ง เพราะอยู่ในระบบของคำ สัญลักษณ์ที่สั้นกระชับ ชัดเจน สะดวกต่อการทำความเข้าใจ และการแปลงไปสู่หน่วยอื่น ๆ ได้



code

noun • UK  /kəʊd/ US  /koʊd/

code noun (LANGUAGE)

★ B2 [C or U] **a system of words, letters, or signs used to represent a message in secret form, or a system of numbers, letters, or signals used to represent something in a shorter or more convenient form:**

*The message was written **in** code.*

*She managed to **decipher/break/crack** (= succeed in understanding) the code.*

Each entry in this dictionary has a grammar code.

Coding - Debug

ADA LOVELACE

FIRST COMPUTER PROGRAMMER

The Analytical Engine

Lovelace's program turned a complex formula into simple calculations that could be encoded on punched cards and fed into Charles Babbage's Analytical Engine, a mechanical computer that he designed but never built. She published it in 1843, a century before the modern computer age.

"I want to put in something about Bernoulli's Number, in one of my Notes, as an example of how an explicit function may be worked out by the engine, without having been worked out by human head and hands first."

$$\frac{x}{e^x - 1} = \frac{1}{1 + \frac{x}{2} + \frac{x^2}{2 \cdot 3} + \frac{x^3}{2 \cdot 3 \cdot 4} + \&c.}$$

A Universal Computer

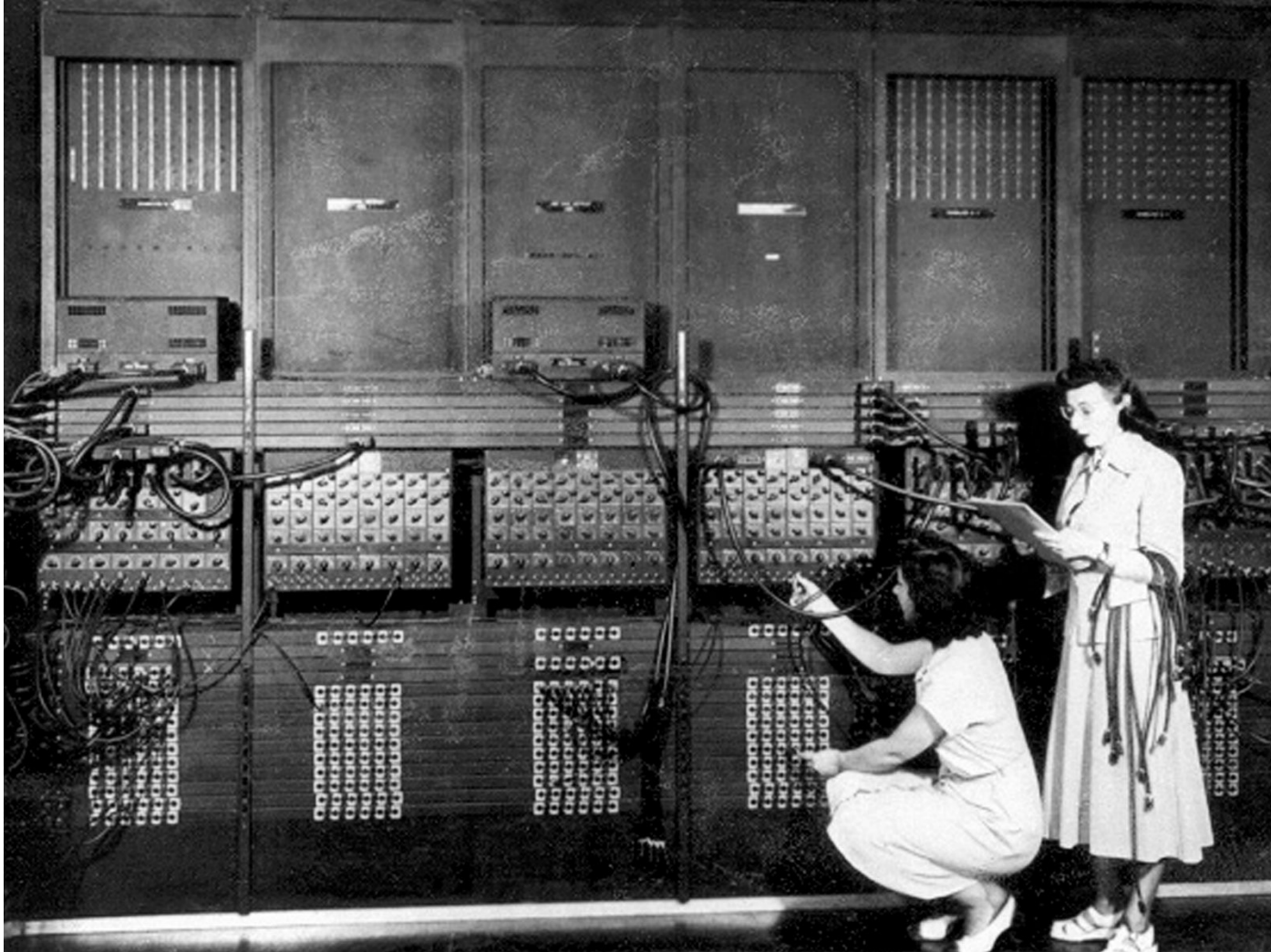
Lovelace did more than write the first computer program. She was also the first person to realise that a general purpose computer could do anything, given the right data and instructions.

"The Analytical Engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves."

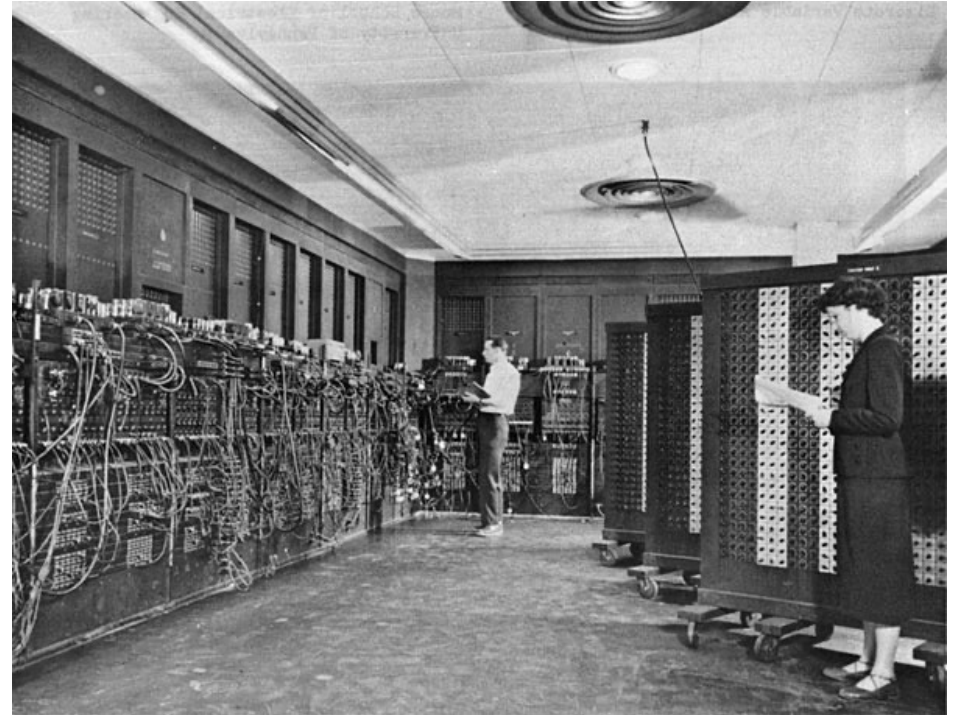
"Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."



Coding - Debug



**Computer Programming and
Developing Applications for Education**
Asst.Prof.Dr.Nutthapat Kaewrattanapat





On the 9th of September of 1946, a Harvard technical team was asked to take a look at a malfunction that was happening with the Mark II calculator.

Coding - Debug

9/9

0800 Andean started

1000 " stopped - andean ✓

1300 (032) MP-MC 1.582147000
2.130476415 (23) 4.615925059 (-2)

(033) PRO 2 2.130476415
2.130676415

concord

Relays 6-2 in 033 failed special speed test
in relay " 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi Adder Test.

1545 Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630 Andean started.

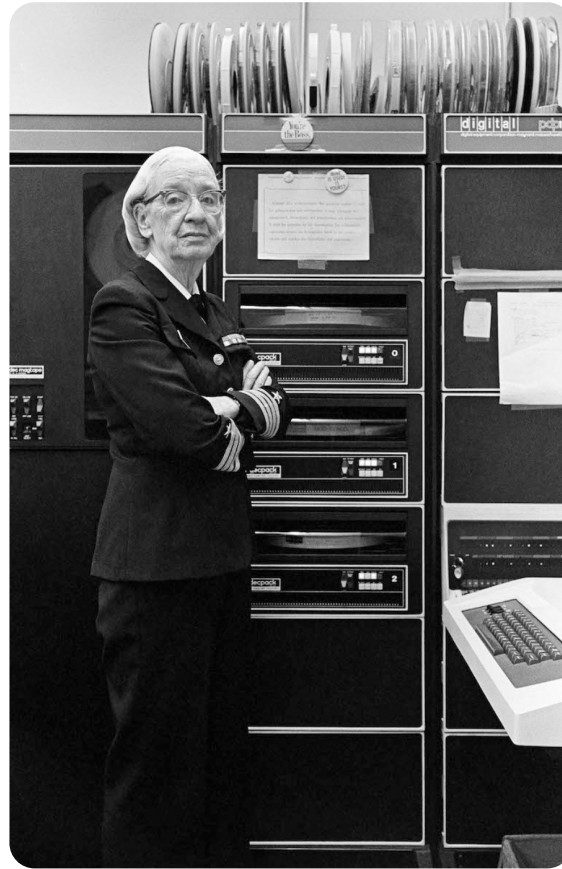
1700 closed down.

Relay 33
2145
Relay 33



Grace Hopper, an American computer scientist and States Navy rear admiral working on the Mark II calculator at the time, added the “First actual case of bug found,” and that’s the first anyone used the word bug to describe a computer glitch. Today, the term debugging is standard.

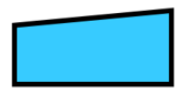
Grace Brewster Hopper



**Computer Programming and
Developing Applications for Education**
Asst.Prof.Dr.Nutthapat Kaewrattanakat

โค้ดดิ้ง (Coding)

รหัสจำลองสำหรับการเดินทาง	
รหัสจำลอง	ความหมาย
→	เดินไปข้างหน้า
↑	เดินขึ้นข้างบน
↓	เดินลงไปข้างล่าง
←	เดินไปทางซ้าย
↗	เดินเฉียงขวา
↖	เดินเฉียงซ้าย
↗↖	เดินเลี้ยวขึ้นข้างบน
↘↖	เดินเลี้ยวลงข้างล่าง

สัญลักษณ์	ความหมาย	ภาษาอังกฤษ
	หมายถึง จุดเริ่มต้น หรือ จุดสิ้นสุดการทำงาน	Begin และ End
	หมายถึง ทิศทางการไหลของงาน	Flow line
	หมายถึง การดำเนินการ หรือ การประมวลผล	Process
	หมายถึง การนำเข้าข้อมูลด้วย การป้อนค่าทางแป้นพิมพ์	Manual input
	หมายถึง การตัดสินใจตาม เงื่อนไขที่กำหนดไว้	Decision
	หมายถึง การแสดงผลบนหน้าจอ	Display

โค้ดดิ้ง (Coding)

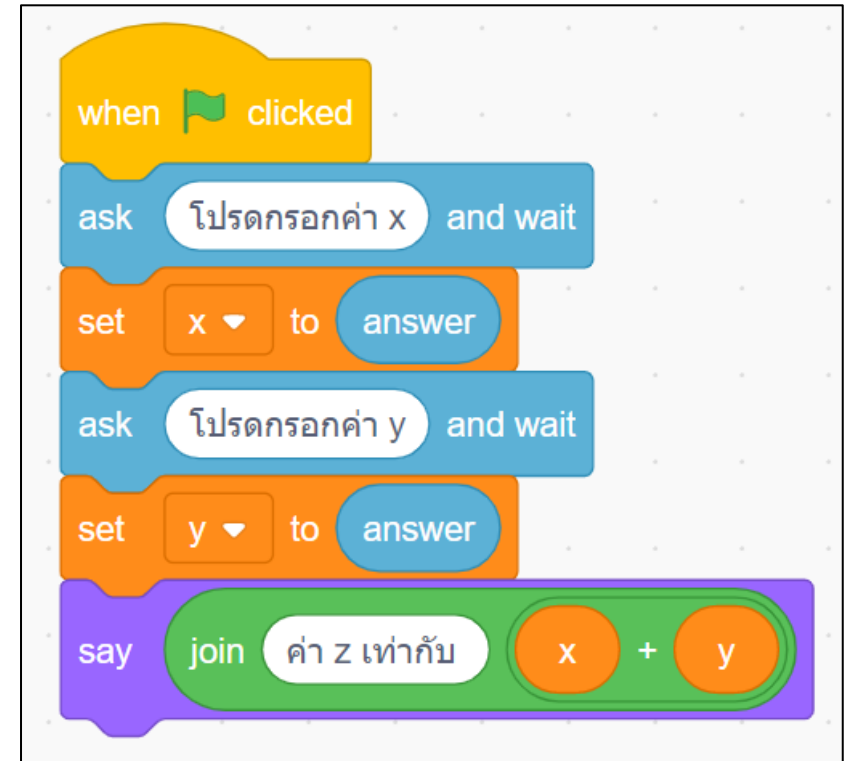
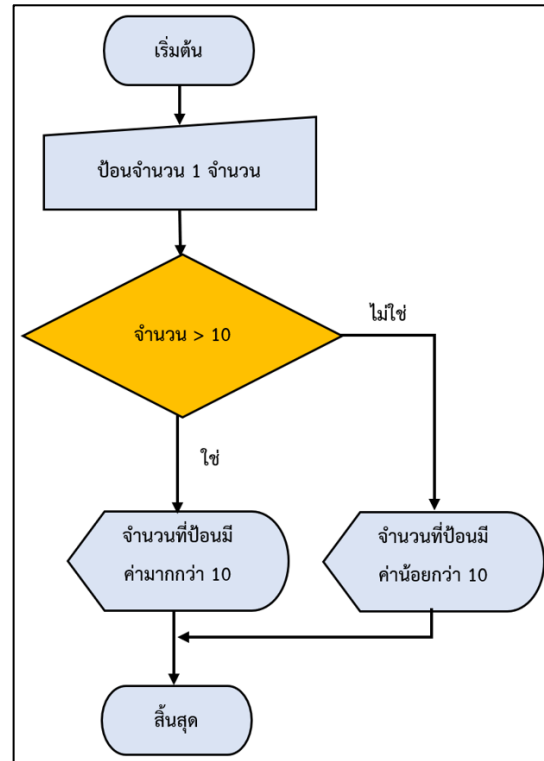
รหัสจำลองของชุดคำสั่งที่ 1 : ลากเส้นจากจุด

(C,2) → (A,4) → (A,7) → (B,7) →
(B,5) → (C,5) → (C,9) → (D,9) →
(D,7) → (G,7) → (G,9) → (H,9) →
(H,6) → (I,7) → (I,6) → (H,5) → (H,2)
→ (C,2)

รหัสจำลองของชุดคำสั่งที่ 2 : ลากเส้นจากจุด

(D,5) → (E,6) → (F,6) → (F,3) →
(E,3) → (D,4)

แสดงผล: ?



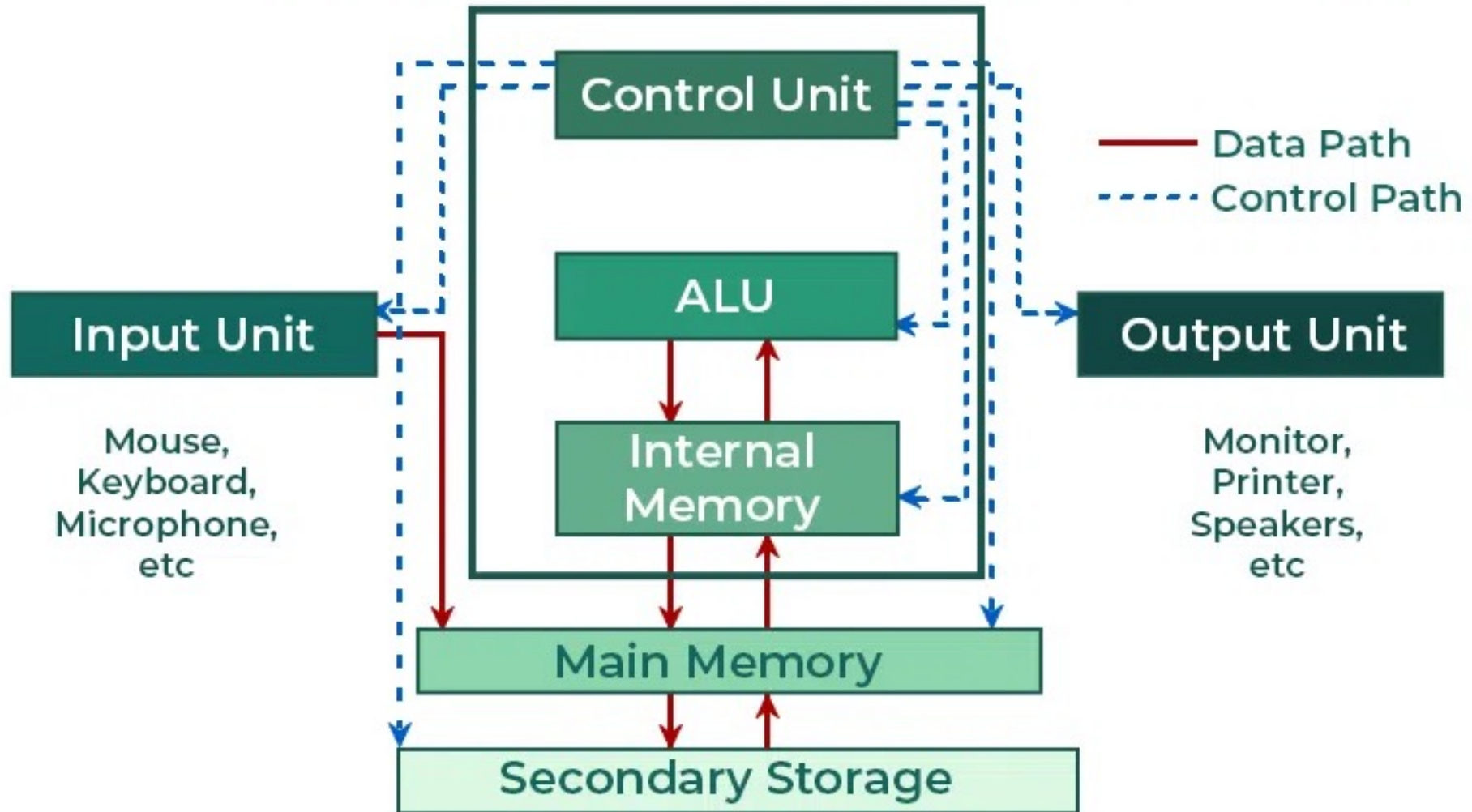
Computer System

Computer

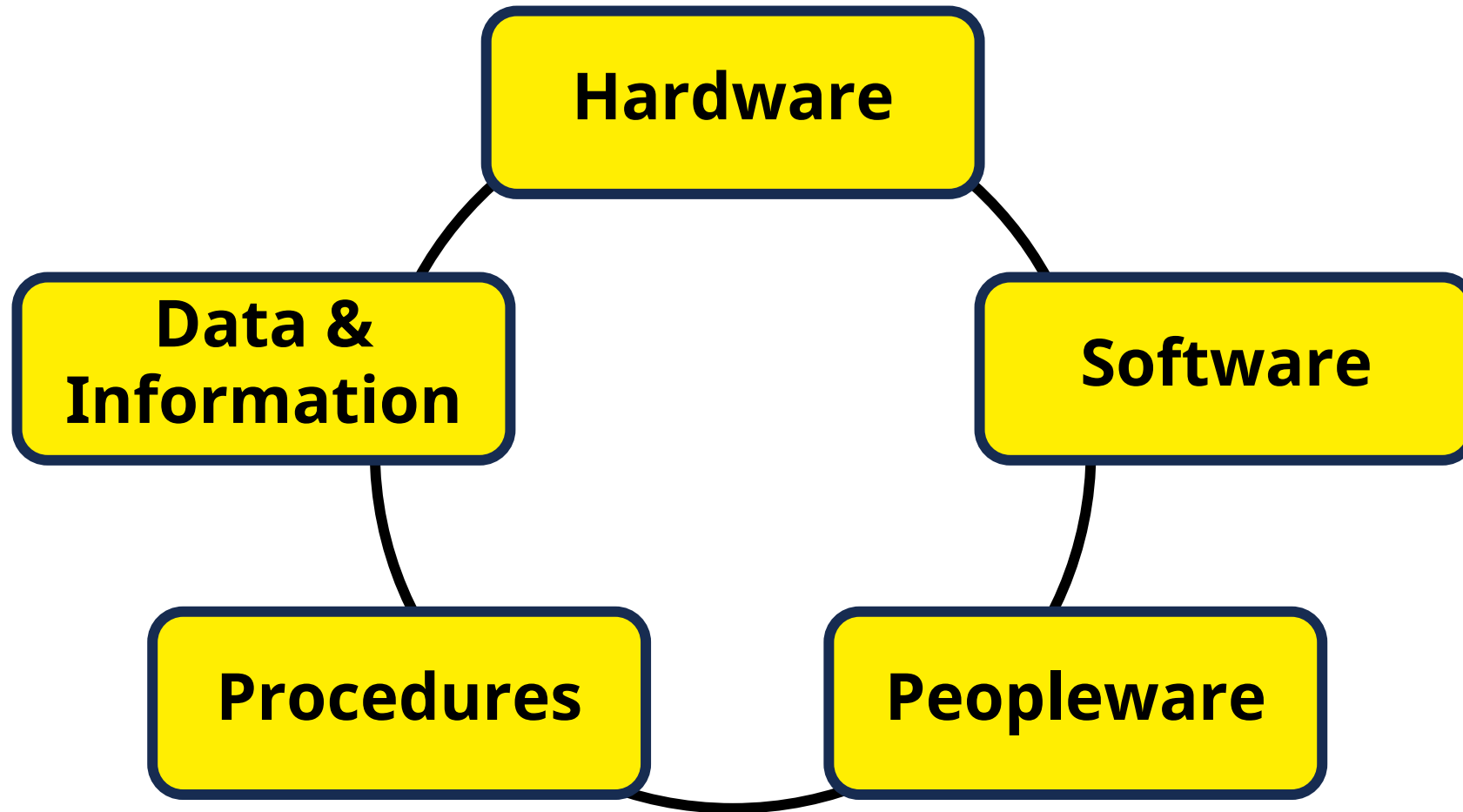


ComputerHope.com

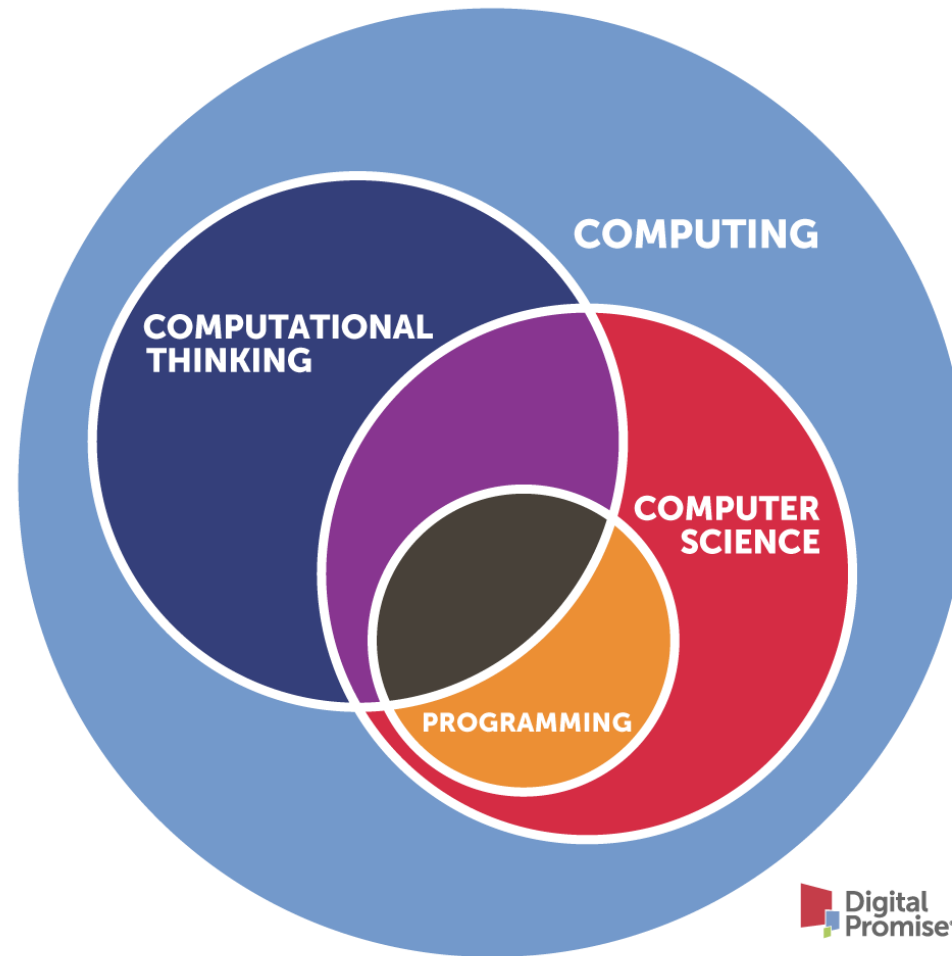
Computer Components



Computer Components

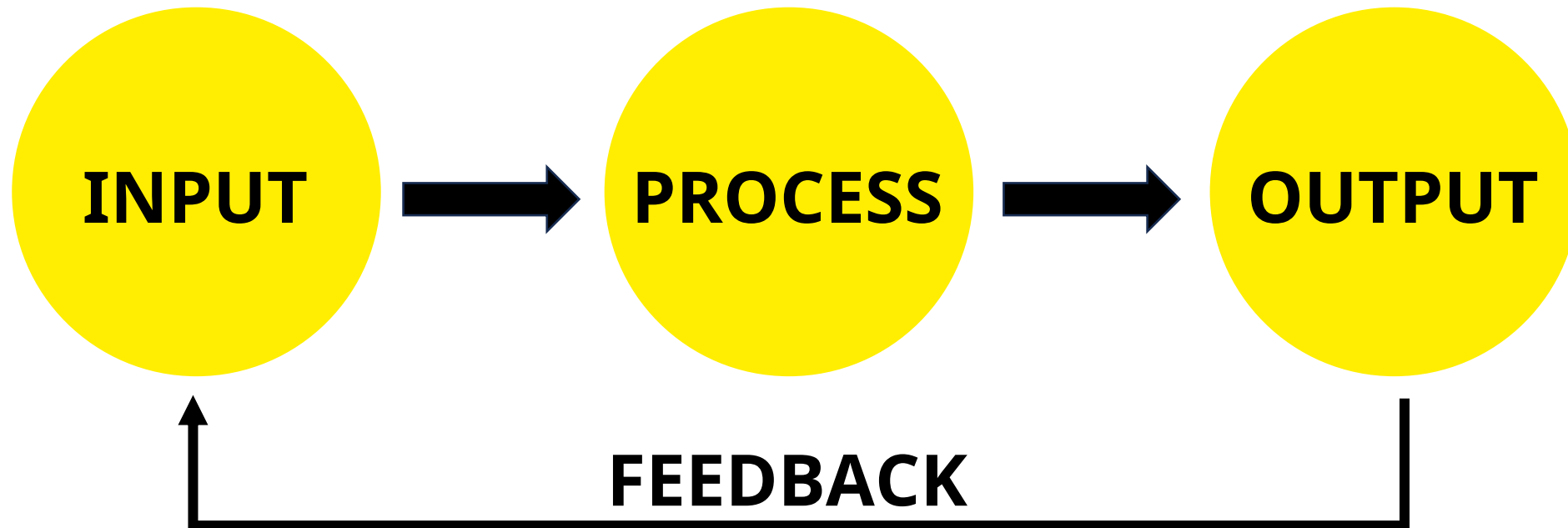


The relationship between computer science (CS), computational thinking (CT), programming and computing.

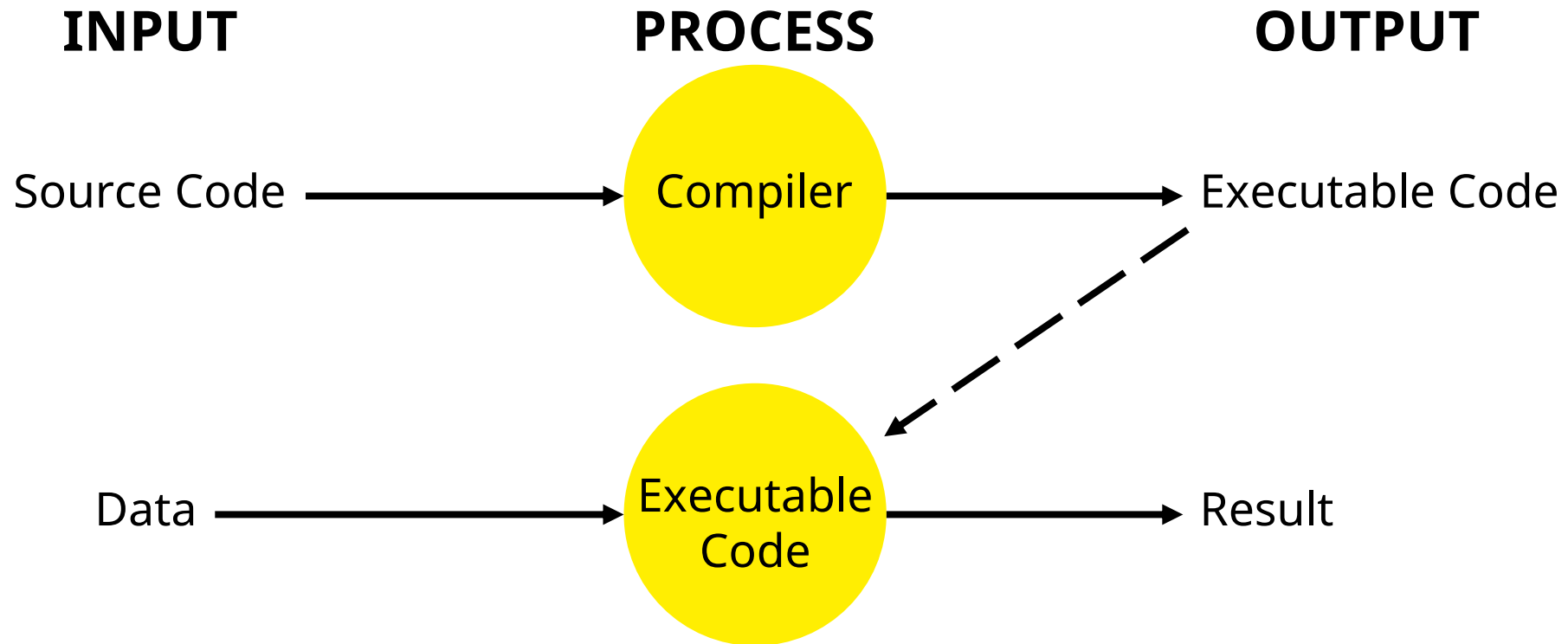


อ้างอิง:
<https://digitalpromise.org/initiative/computational-thinking/computational-thinking-for-next-generation-science/what-is-computational-thinking/>

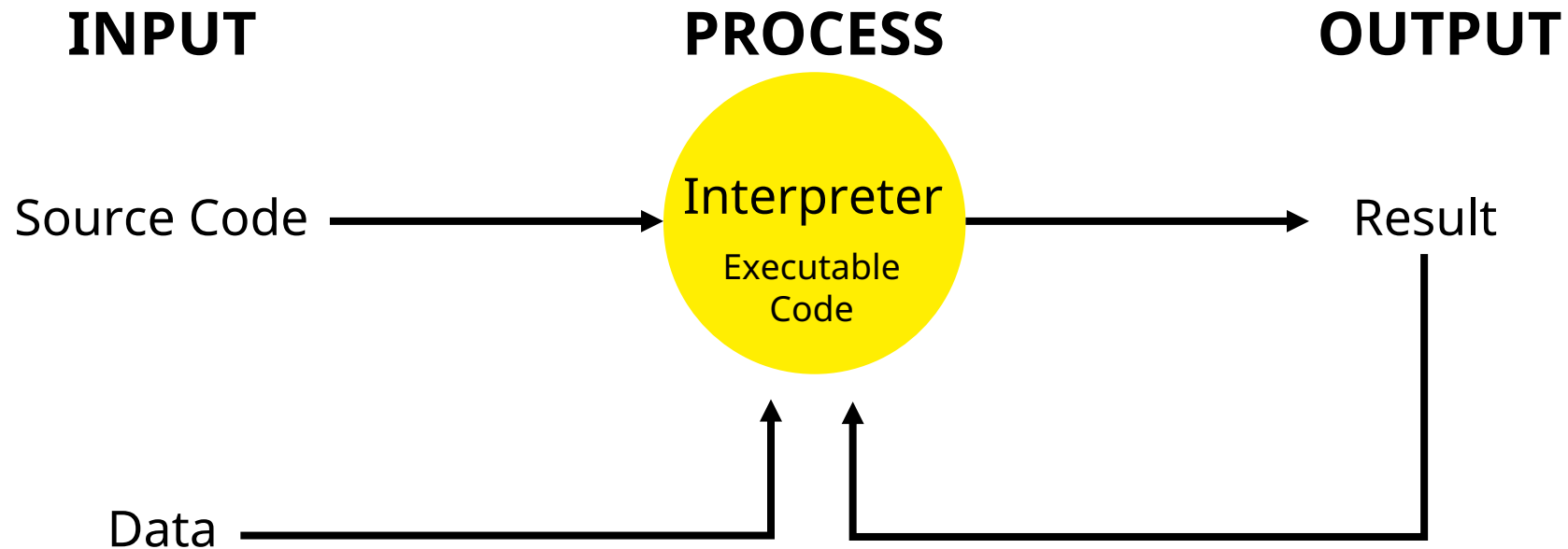
System Theory



Compiler



Interpreter – Line Interpretation



Difference Between High-Level Language & Low-Level Language

High-Level Language

Python, C, C++, C# Language

Procedural Language

Object Oriented: OO

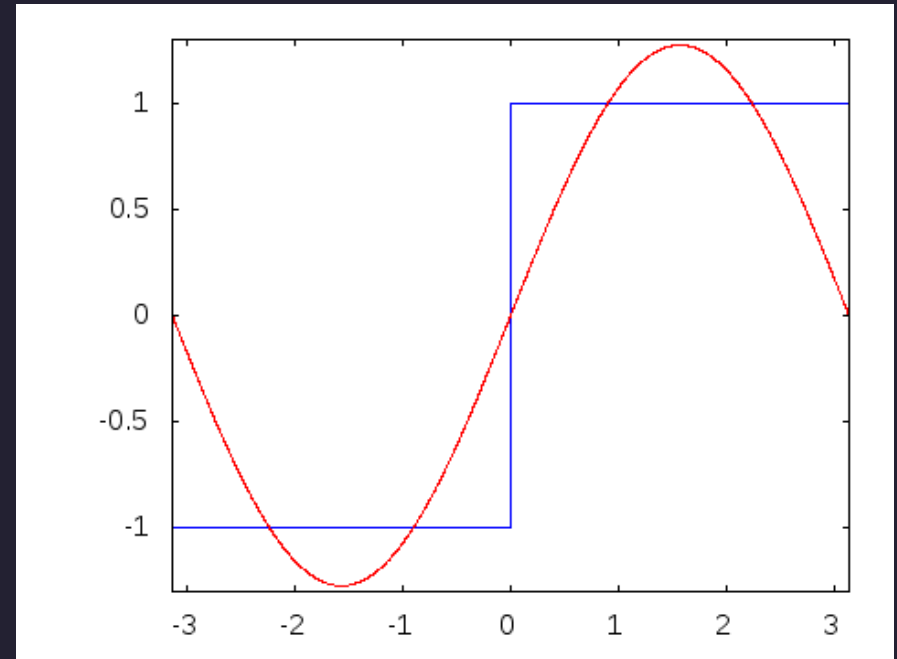
Low-Level Language

Assembly Language

Machine Language



Hardware



Difference Between High-Level Language & Low-Level Language

High-Level Language

Python, C, C++, C# Language

Procedural Language

Object Oriented: OO

Low-Level Language

Assembly Language

Machine Language

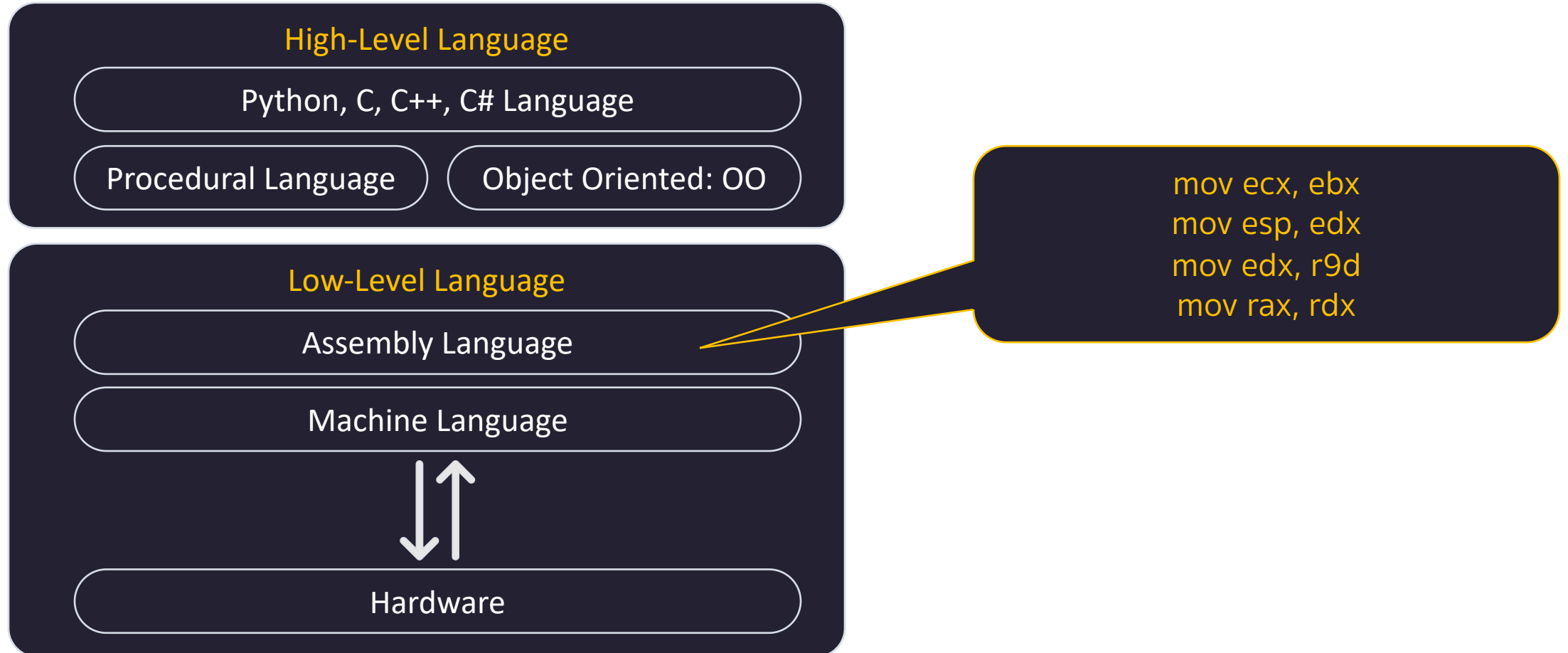


Hardware

1001001001010111101010101



Difference Between High-Level Language & Low-Level Language



Difference Between High-Level Language & Low-Level Language

High-Level Language

Python, C, C++, C# Language

Procedural Language

Object Oriented: OO

```
price = 25000
product = "iPad"
print("%s price:%.2f" %(product, price))
```

Low-Level Language

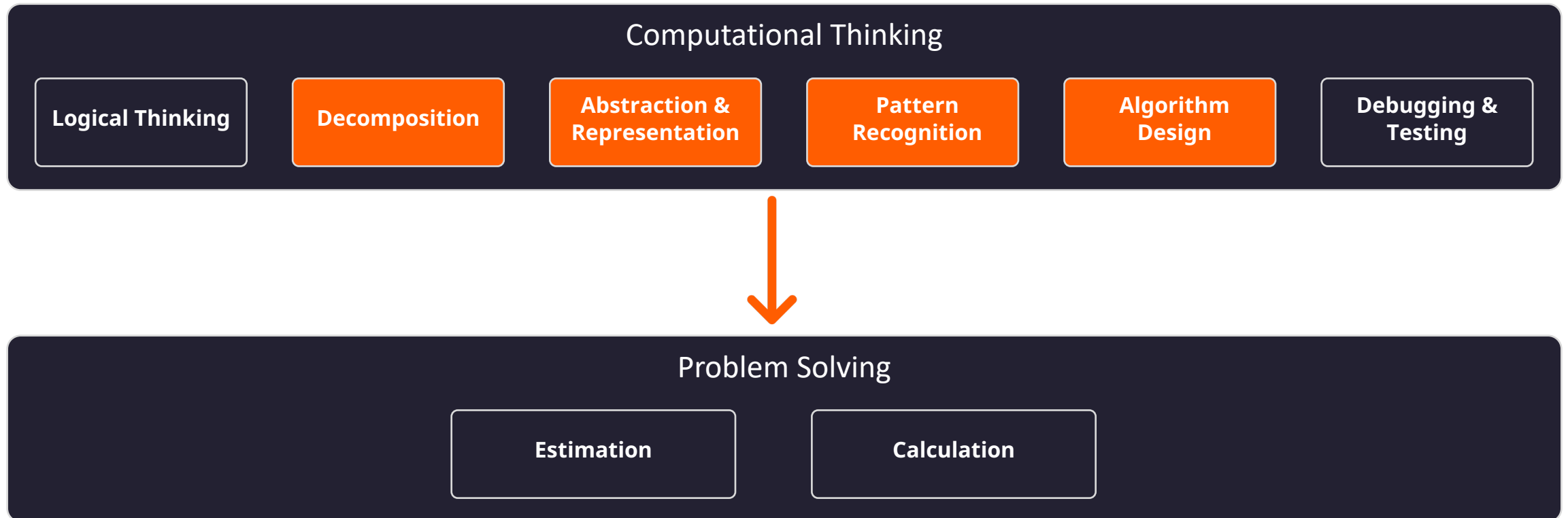
Assembly Language

Machine Language

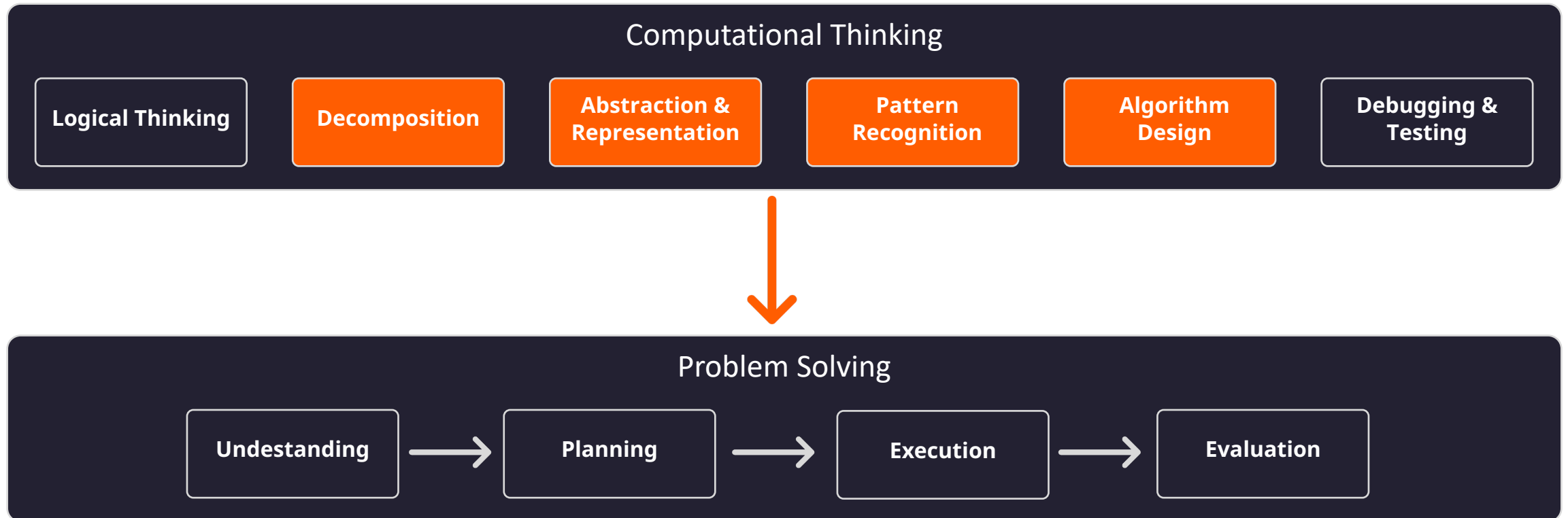


Hardware

Computational Thinking & Problem Solving

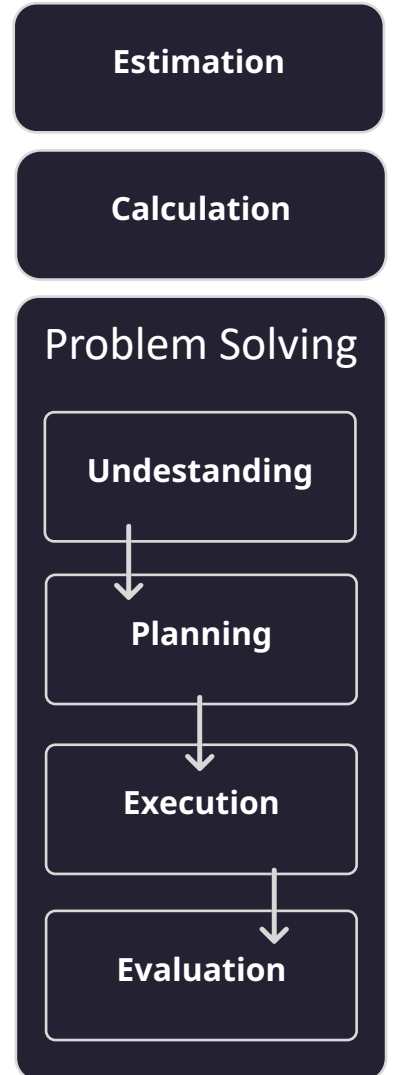
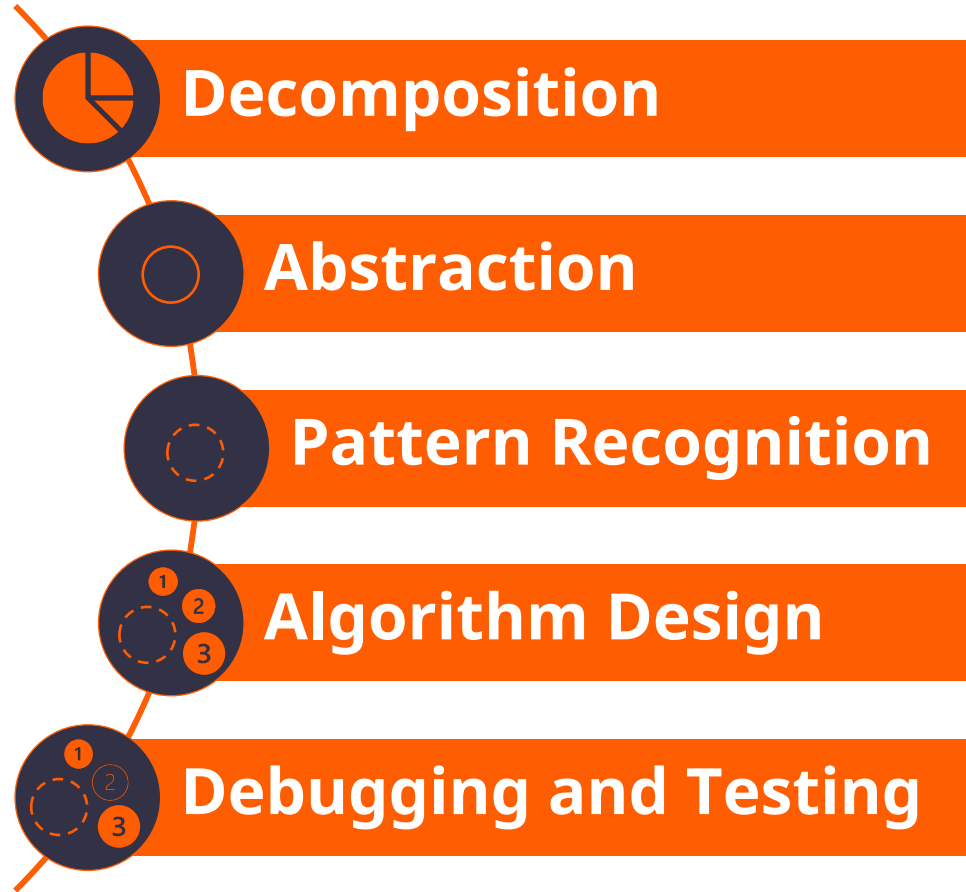


Computational Thinking & Problem Solving



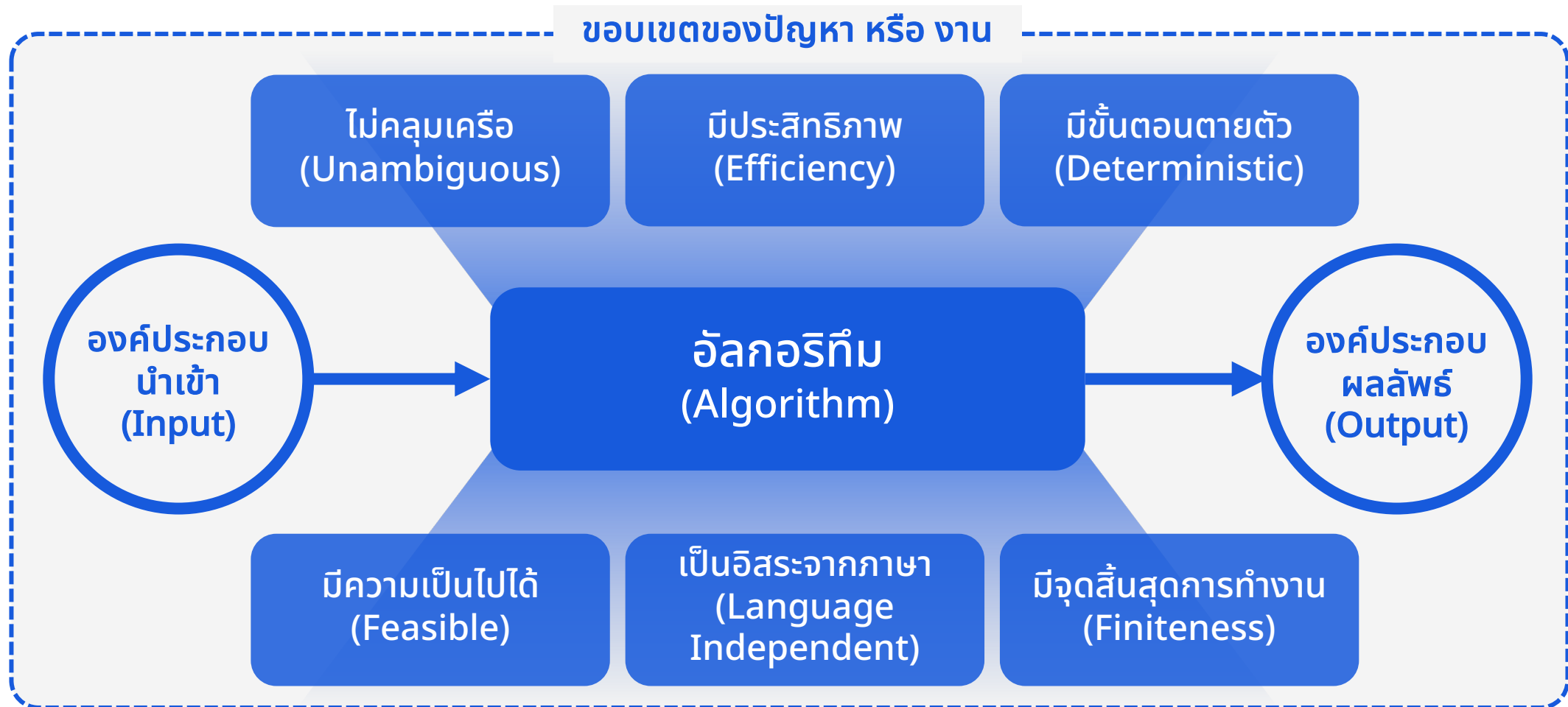
Python & Computational Thinking

- Computational thinking is a problem-solving method that involves
 - breaking down complex problems into smaller,
 - more manageable parts,
 - and using logical reasoning and algorithms to solve them.
- Python is a programming language that can be used to implement computational thinking.



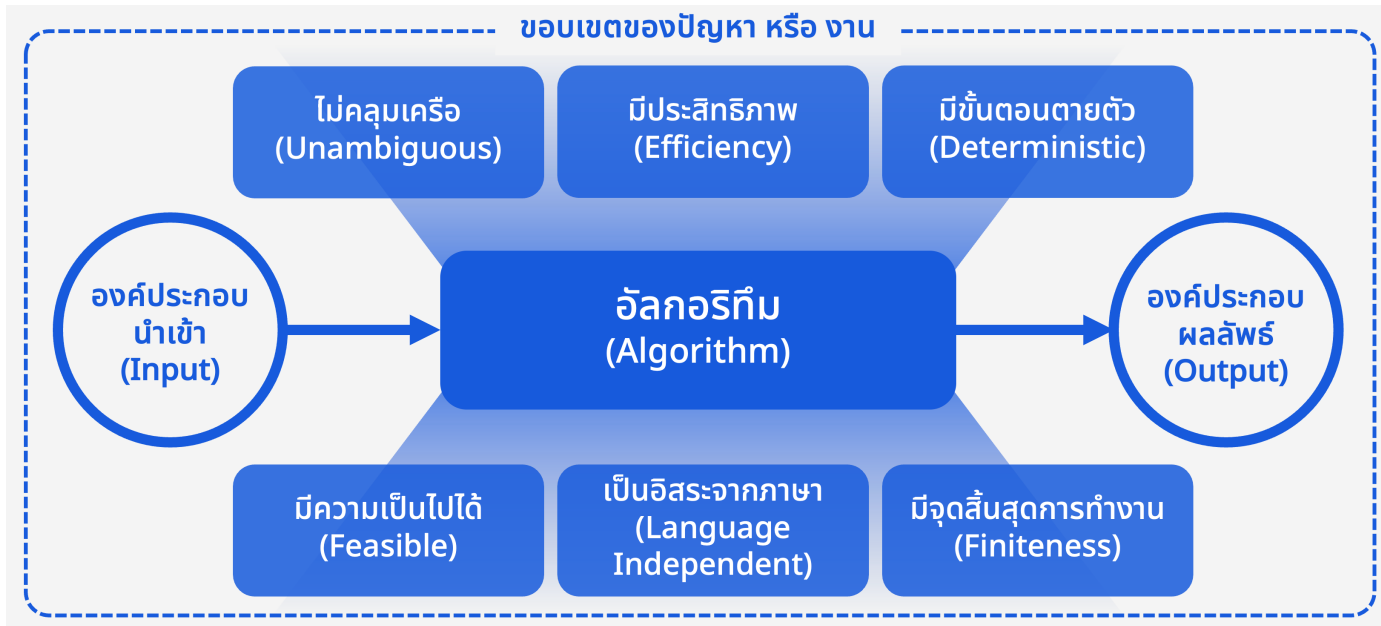
คุณลักษณะของอัลกอริทึม

Characteristics of an Algorithm



คุณลักษณะของอัลกอริทึม

Characteristics of an Algorithm



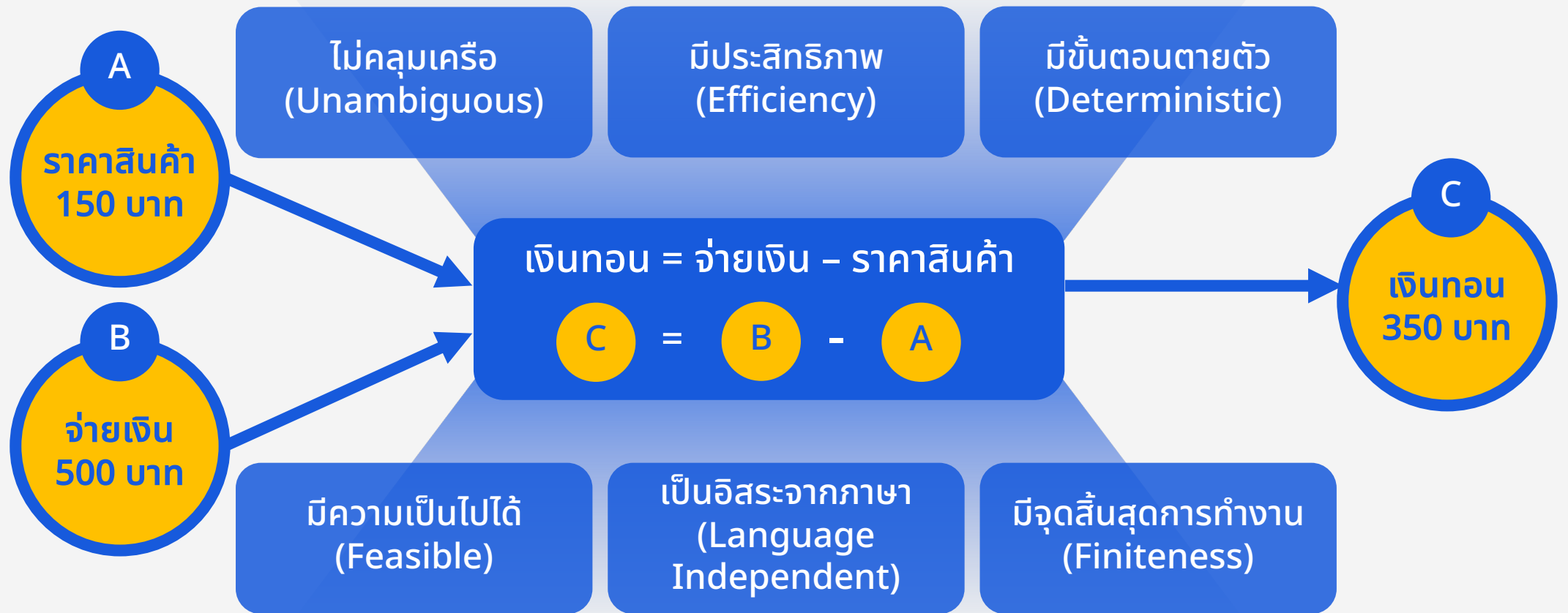
อ้างอิง

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press. ISBN: 978-0262033848.
2. Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley Professional. ISBN: 978-0321573513.

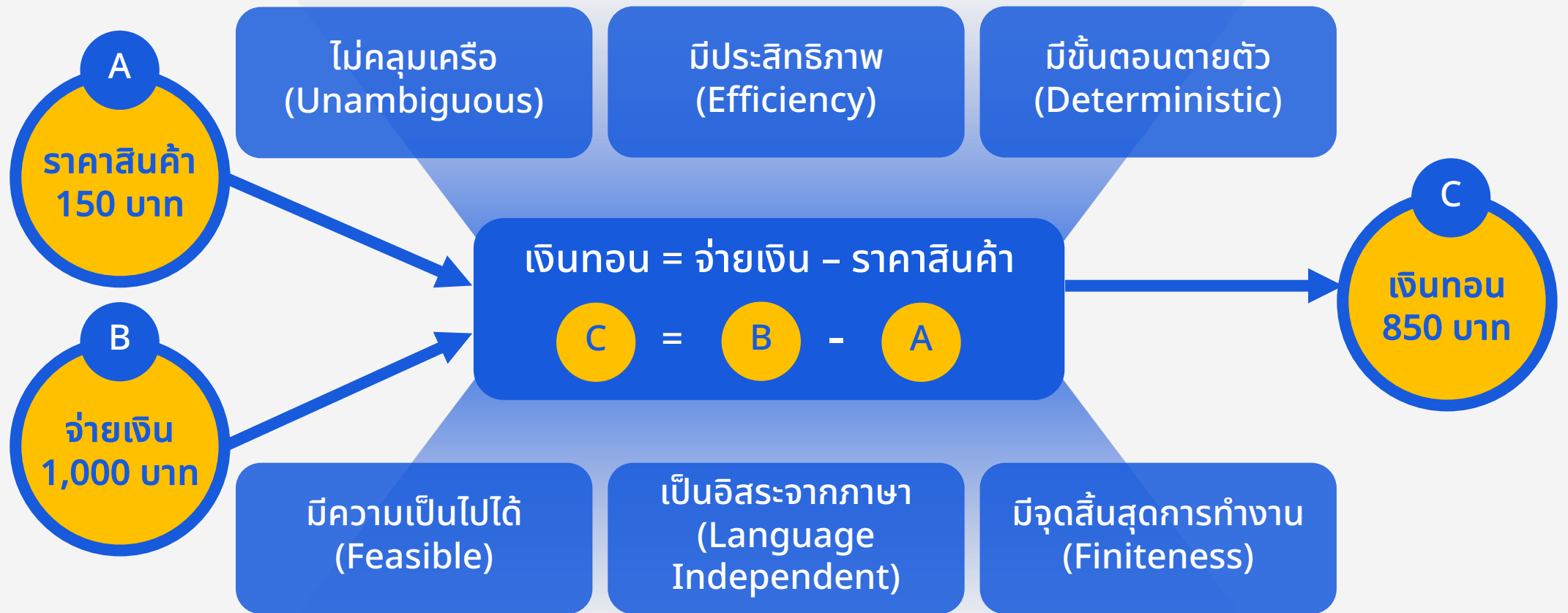
ลักษณะของอัลกอริทึมที่ดีที่ประกอบด้วยคุณสมบัติ ดังนี้

1. **ไม่คลุมเครือ (Unambiguous)** อัลกอริทึมต้องไม่มีความกำกวมที่ไม่ชัดเจน และต้องมีชุดคำสั่งที่เข้าใจง่ายและไม่ทำให้เกิดความสับสนในการประมวลผล
2. **มีประสิทธิภาพ (Efficiency)** อัลกอริทึมควรมีประสิทธิภาพในการทำงาน คือต้องให้ผลลัพธ์ที่ถูกต้องและใช้เวลาน้อยที่สุดในการแก้ปัญหา
3. **มีขั้นตอนตายตัว (Deterministic)** อัลกอริทึมต้องเป็น deterministic ทำความหมายความว่าจะต้องให้ผลลัพธ์ที่เหมือนกันสำหรับข้อมูลนำเข้าที่เหมือนกันทุกครั้งที่รับ
4. **มีความเป็นไปได้ (Feasible)** อัลกอริทึมต้องเป็นไปได้ในการนำไปใช้งานจริง สามารถทำงานบนเครื่องคอมพิวเตอร์หรือสภาพแวดล้อมที่เหมาะสม
5. **เป็นอิสระจากภาษา (Language Independent)** อัลกอริทึมควรไม่เกี่ยวข้องกับภาษาเฉพาะเสียเพียงอย่างเดียว และสามารถนำไปใช้งานในหลายภาษาโปรแกรมได้
6. **มีจุดสิ้นสุดการทำงาน (Finiteness)** อัลกอริทึมต้องสามารถสิ้นสุดการทำงานหลังจากทำขั้นตอนทุกขั้นตอนเสร็จสมบูรณ์ และไม่ต้องเข้าสู่วงจรไม่สิ้นสุด (Infinity Loop)

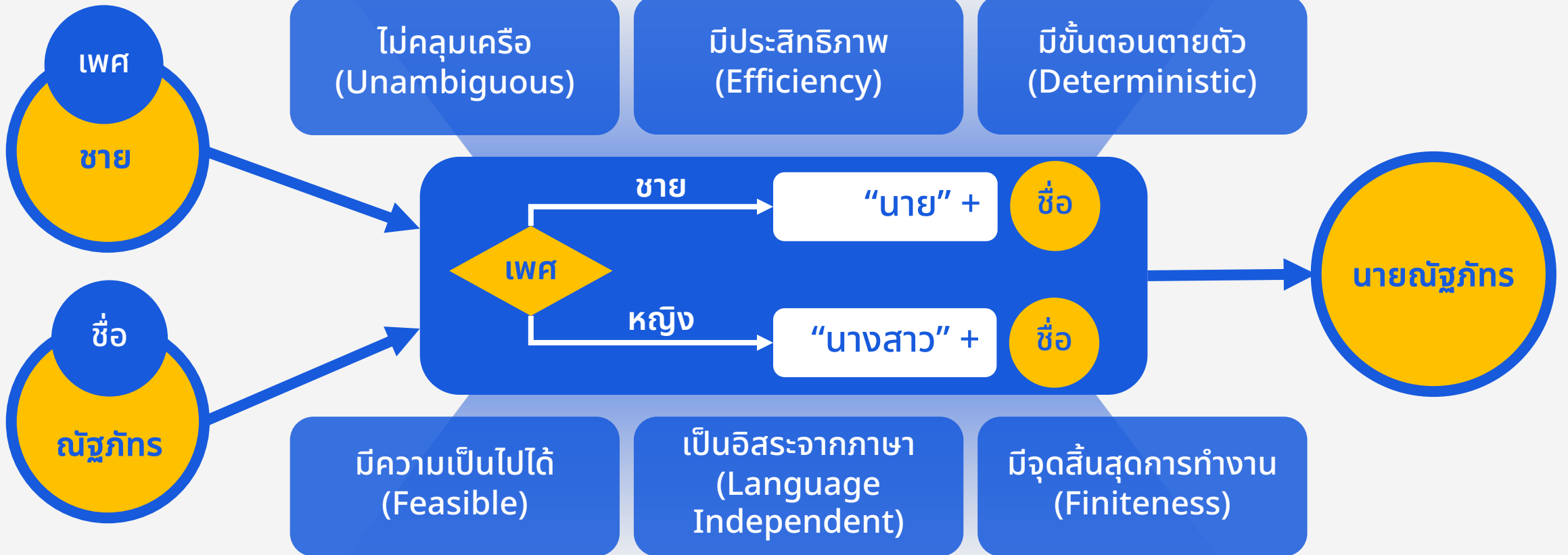
ขอบเขตของปัญหา หรือ งาน
คำนวณเงินทอน



ขอบเขตของปัญหา หรือ งาน
คำนวณเงินทอน

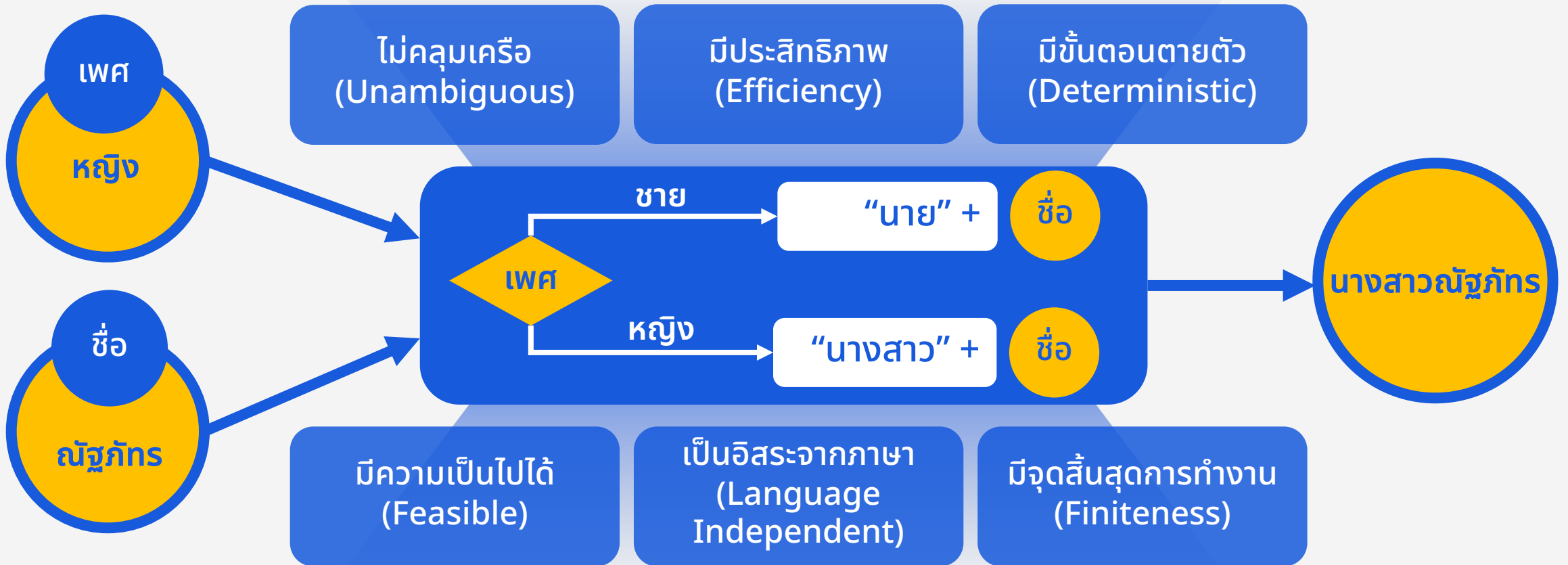


ขอบเขตของปัญหา หรือ งาน
เต็มคำนำหน้าชื่อ



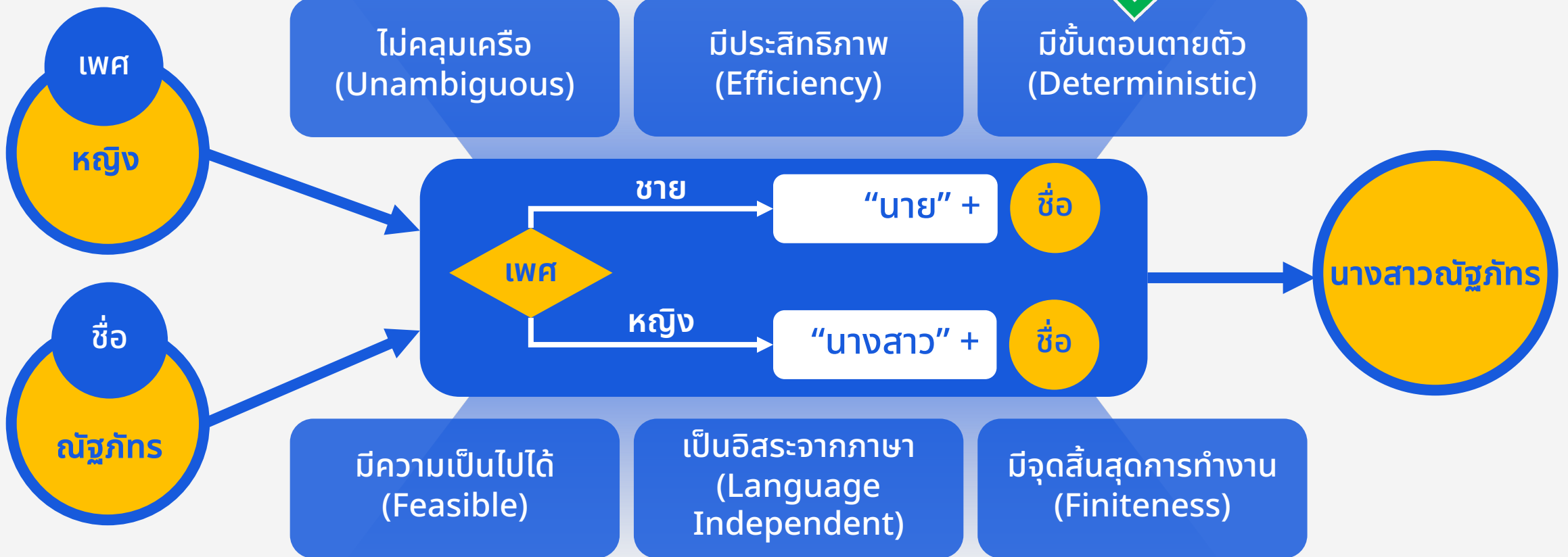
ให้ผลลัพธ์ถูกต้องตามเงื่อนไขที่กำหนดไว้เสมอ คือ คุณลักษณะข้อใด

ขอบเขตของปัญหา หรือ งาน
เต็มคำนำหน้าชื่อ



ให้ผลลัพธ์ถูกต้องตามเงื่อนไขที่กำหนดไว้เสมอ คือ คุณลักษณะข้อใด

ขอบเขตของปัญหา หรือ งาน
เต็มคำนำหน้าชื่อ



การสอน Coding สำหรับผู้เริ่มต้น

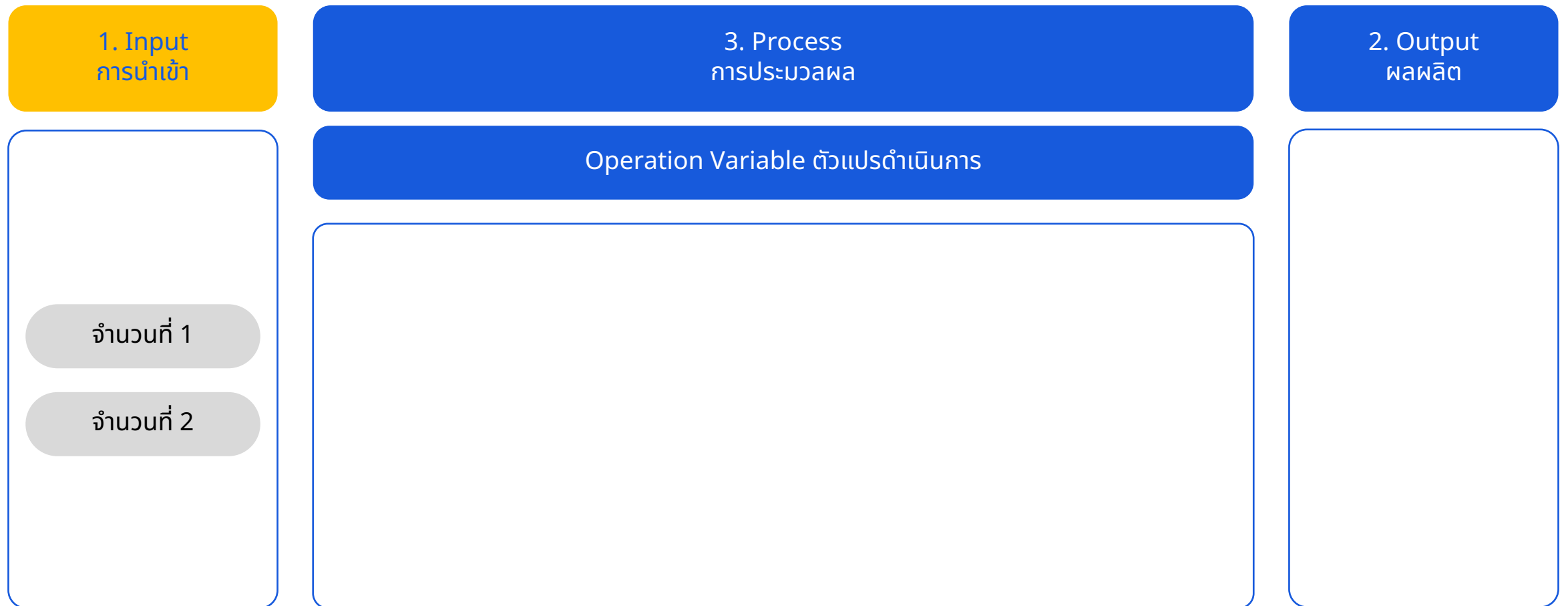
1. Input
การนำเข้า

3. Process
การประมวลผล

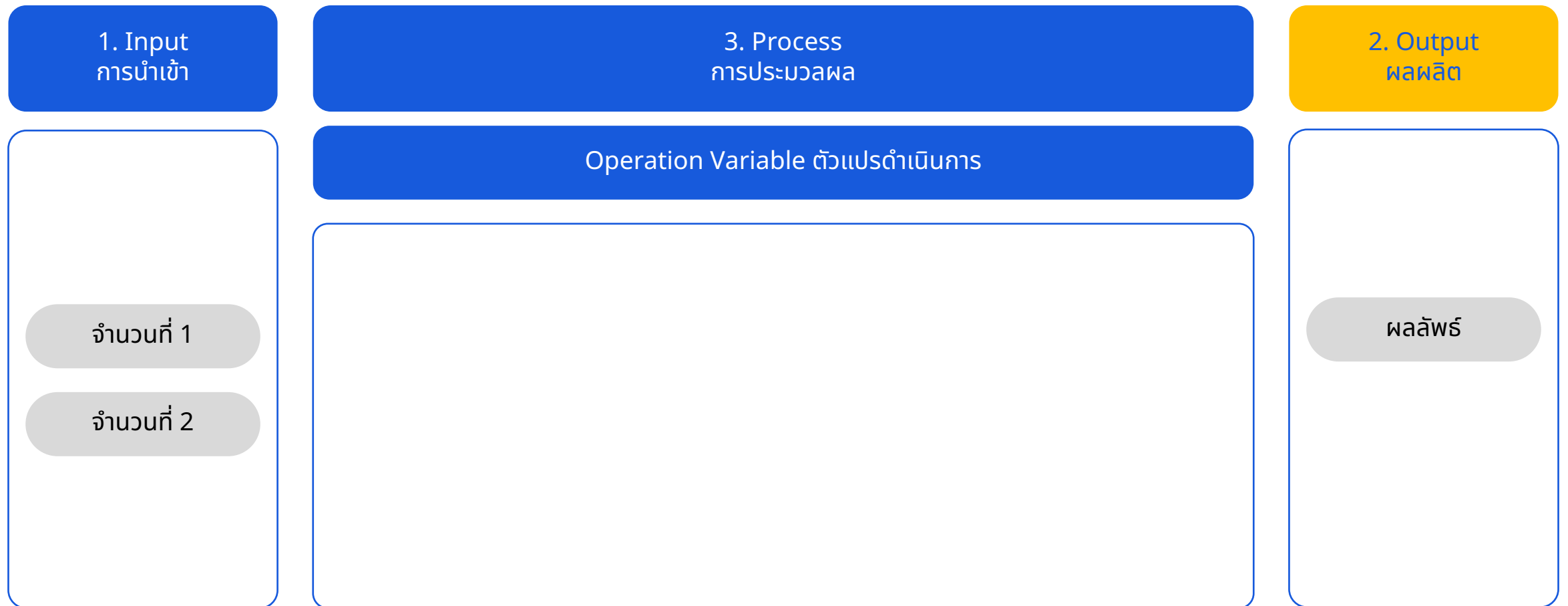
2. Output
ผลผลิต

Operation Variable ตัวแปรดำเนินการ

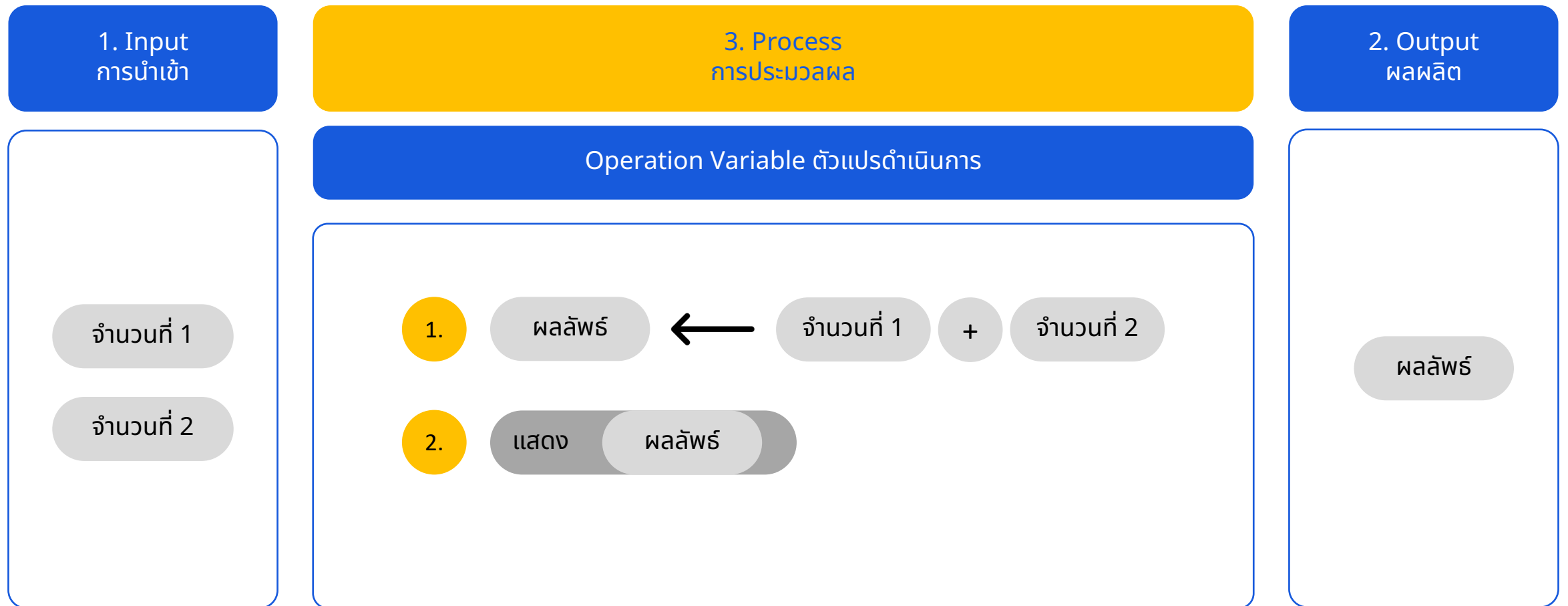
ต้องการผลรวมของค่า 2 จำนวน



ต้องการผลรวมของค่า 2 จำนวน

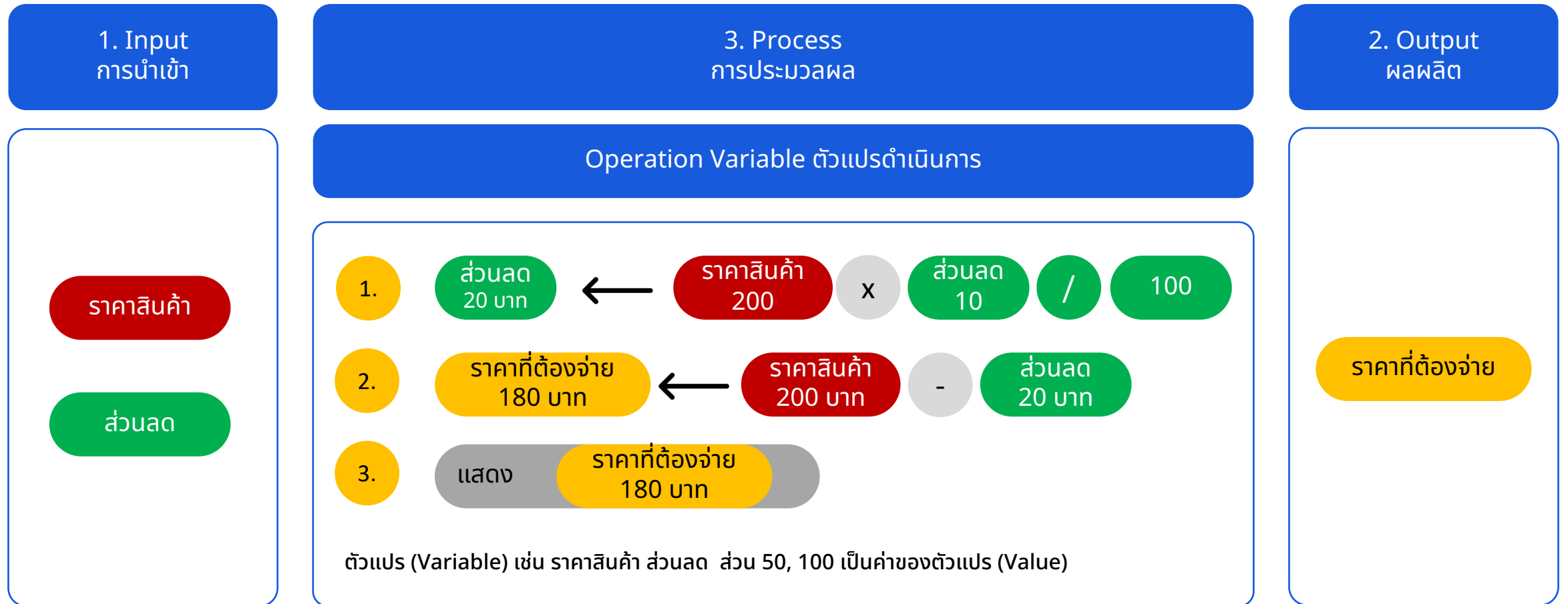


ต้องการผลรวมของค่า 2 จำนวน



ต้องการคำนวณส่วนลดราคาสินค้า

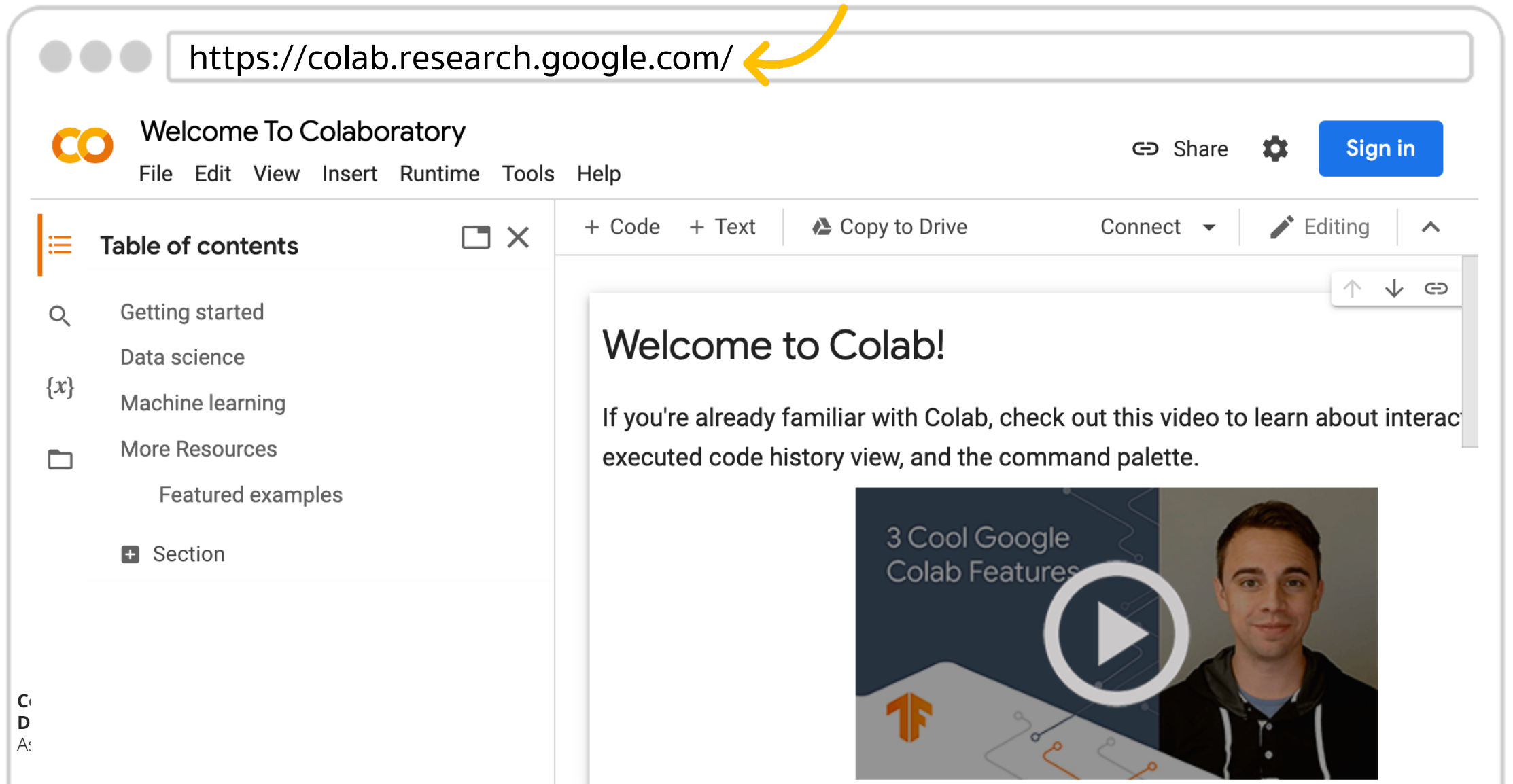
ราคาสินค้า 200 บาท ส่วนลด 10% ต้องจ่ายกี่บาท



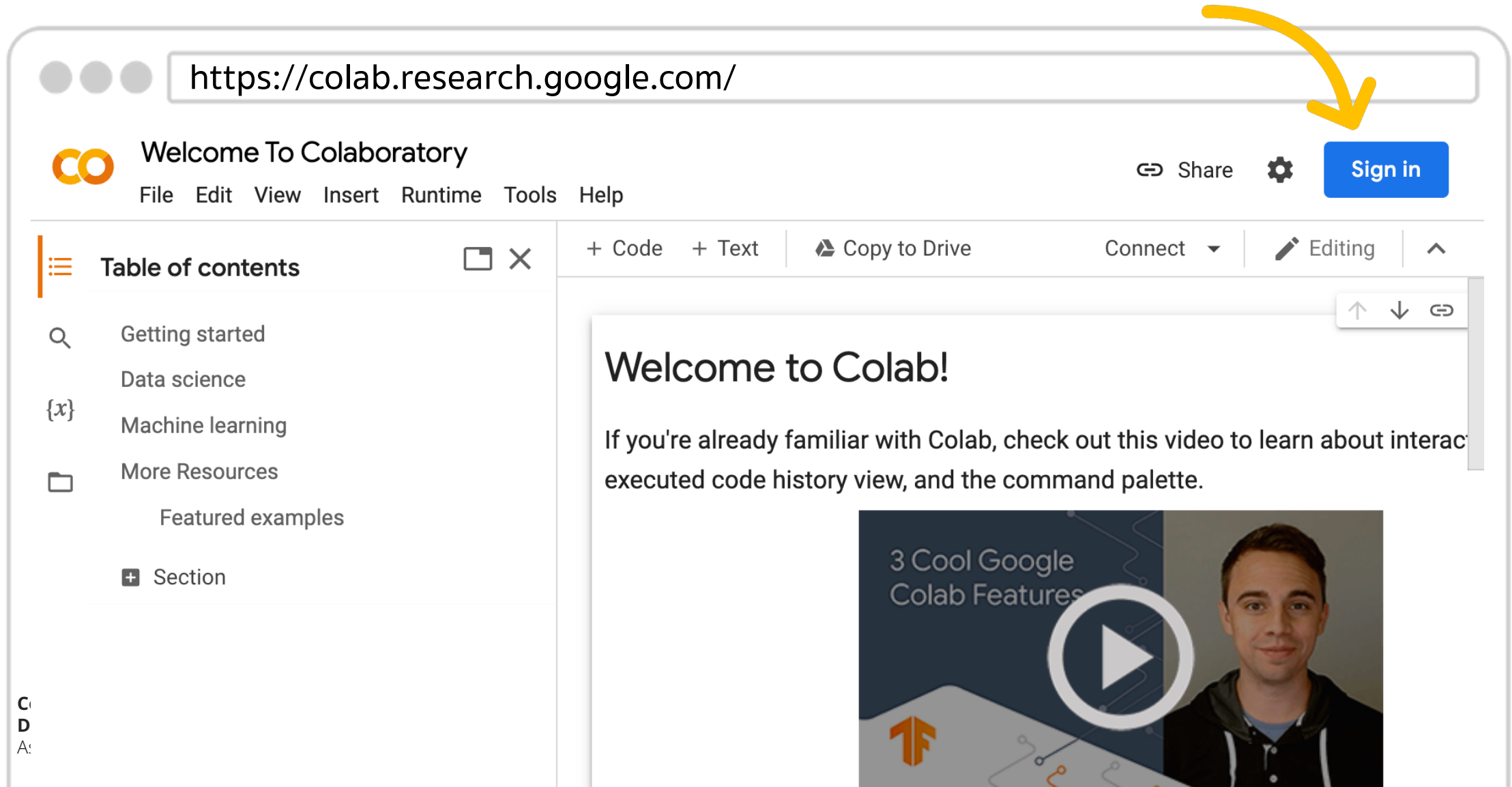
Python Programming Steps Using Google Colab

1. Go to <https://colab.research.google.com/>
2. Log in with your gmail
3. File -> New Notebook
4. Coding with python commands on cell
5. Run cell

1. Go to <https://colab.research.google.com/>



2. Log in with your gmail



The screenshot displays the Google Colaboratory (Colab) web interface. The browser's address bar shows the URL `https://colab.research.google.com/`. The page header includes the Colab logo, the text "Welcome To Colaboratory", and a menu with options: File, Edit, View, Insert, Runtime, Tools, and Help. On the right side of the header, there are links for "Share" and "Settings" (gear icon), and a prominent blue "Sign in" button. A yellow arrow points to this "Sign in" button. Below the header, the left sidebar contains a "Table of contents" section with a search icon and a list of items: "Getting started", "Data science", "Machine learning" (marked with a selection icon {x}), "More Resources", and "Featured examples". A "+ Section" button is at the bottom of the sidebar. The main content area features a "Welcome to Colab!" heading, a paragraph of introductory text, and a video player titled "3 Cool Google Colab Features" with a play button overlay. The video player shows a man's face. At the bottom left, there is a small vertical logo with the letters "C", "D", and "A" stacked.

https://colab.research.google.com/

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Settings Sign in

Table of contents

- Getting started
- Data science
- {x} Machine learning
- More Resources
- Featured examples
- + Section

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interacting with Colab, including the executed code history view, and the command palette.

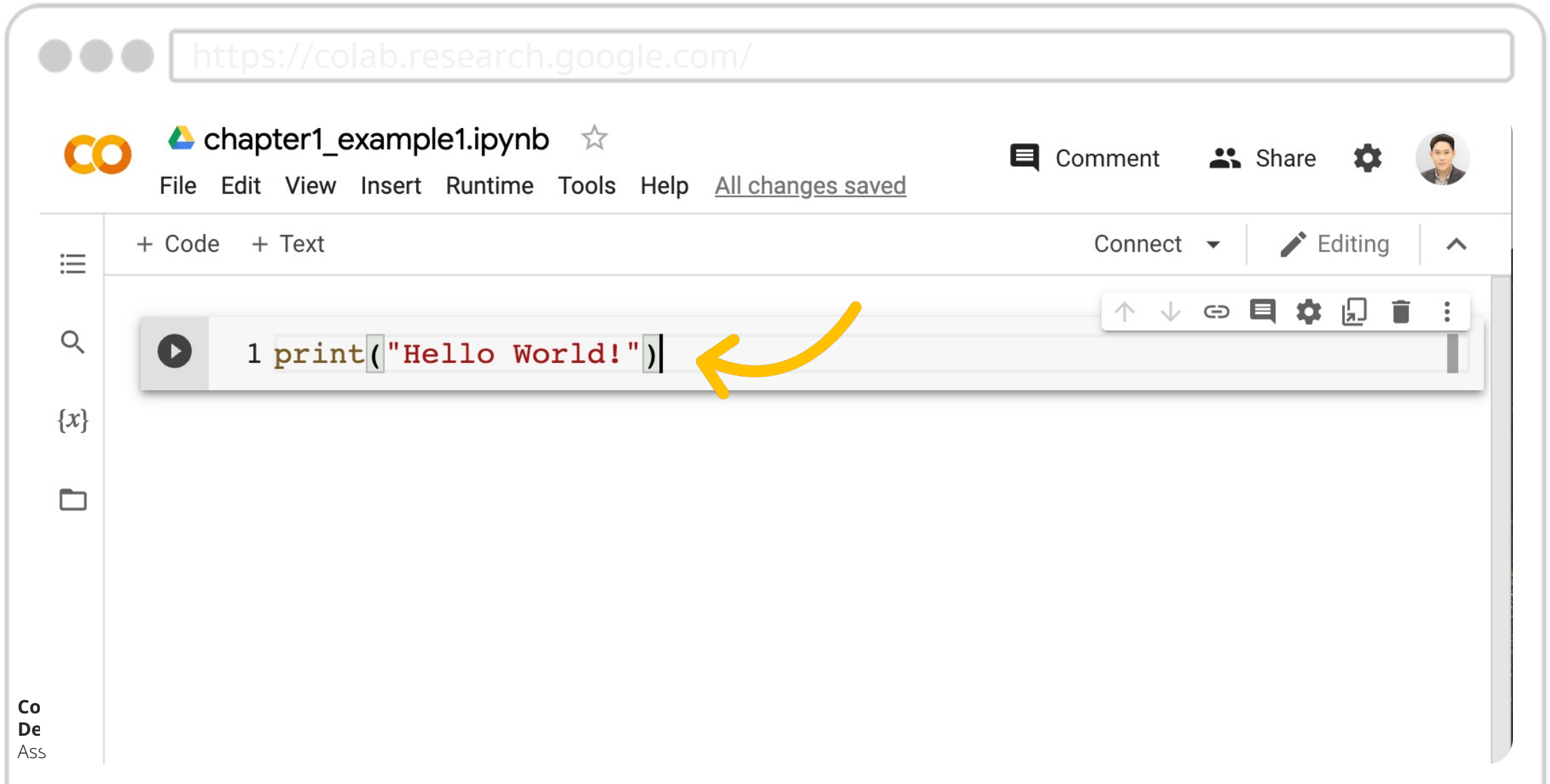
3 Cool Google Colab Features

C
D
A

3. File -> New Notebook

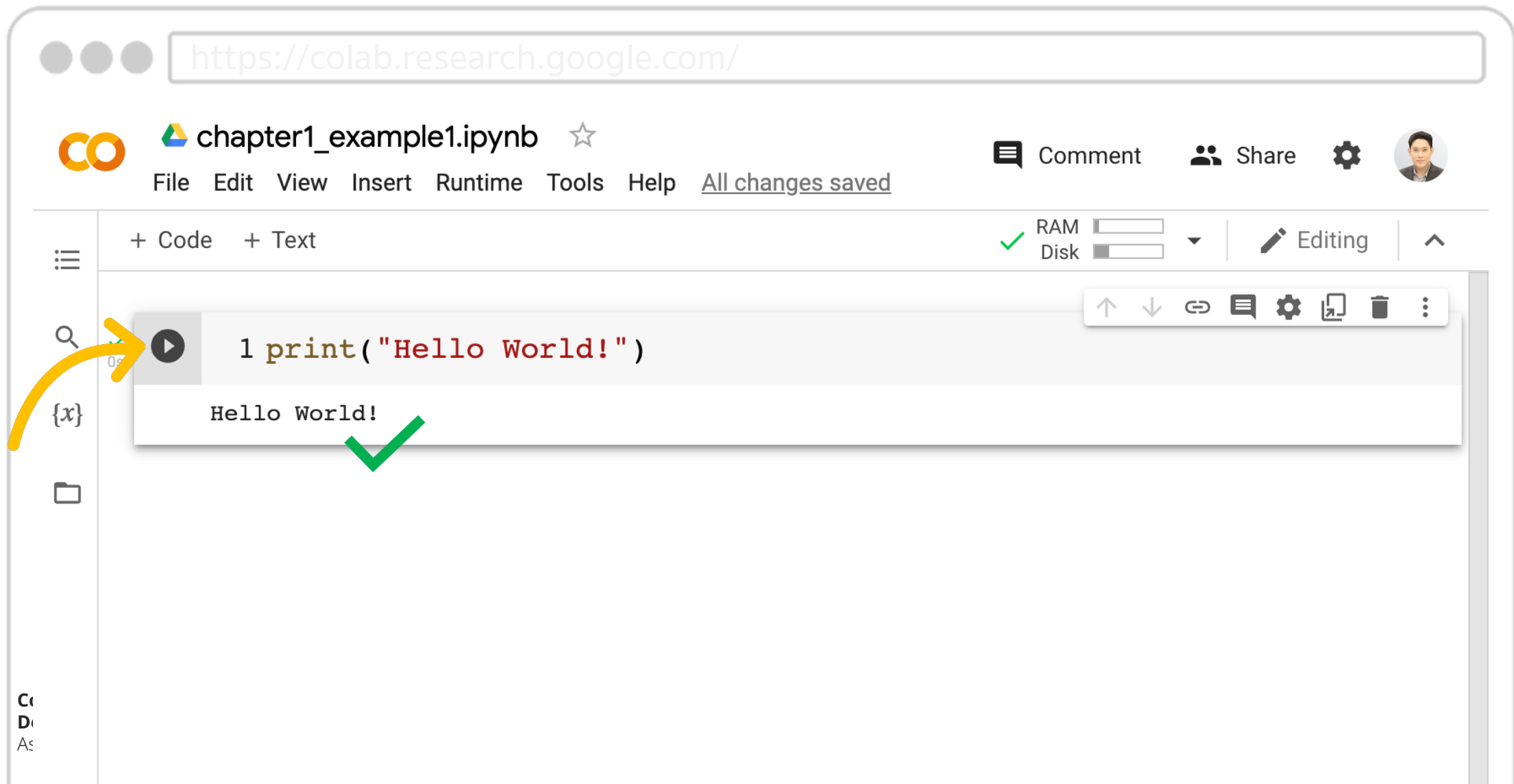
The screenshot shows the Google Colaboratory web interface. The browser address bar displays `https://colab.research.google.com/`. The main header includes the Colab logo, the text "Welcome To Colaboratory", and links for "Share", "Settings", and a user profile. Below the header is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The "File" menu is open, showing options: "New notebook", "Open notebook" (with keyboard shortcut ⌘/Ctrl+O), "Upload notebook", "Rename", "Save a copy in Drive", "Save a copy as a GitHub Gist", "Save a copy in GitHub", "Save" (with keyboard shortcut ⌘/Ctrl+S), "Revision history", "Download", and "Print" (with keyboard shortcut ⌘/Ctrl+P). A yellow circle with the number "1" points to the "File" menu, and a yellow circle with the number "2" points to the "New notebook" option. The main content area displays "Welcome to Colab!" and a video thumbnail titled "3 Cool Google Colab Features" with a play button overlay.

4. Coding with python commands on cell



The screenshot displays the Google Colab web interface. The browser's address bar shows the URL `https://colab.research.google.com/`. The notebook title is `chapter1_example1.ipynb`, and the status indicates `All changes saved`. The top navigation bar includes links for `File`, `Edit`, `View`, `Insert`, `Runtime`, `Tools`, and `Help`, along with buttons for `Comment`, `Share`, and a settings icon. The left sidebar contains icons for a menu, search, and file explorer. The main workspace shows a code cell with the Python command `1 print("Hello World!")`. A yellow arrow points to the closing parenthesis of the command. The cell's toolbar includes icons for running, undo, redo, and other actions. The bottom left corner features the text `Co De Ass`.

5. Run cell



The screenshot shows the Google Colab web interface. The browser address bar displays `https://colab.research.google.com/`. The document title is `chapter1_example1.ipynb`. The top navigation bar includes a menu (File, Edit, View, Insert, Runtime, Tools, Help) and status indicators (Comment, Share, settings, and a user profile). Below the navigation bar, there are tabs for `+ Code` and `+ Text`. On the right side of the code editor, there are status indicators for RAM and Disk usage, a green checkmark, and a button labeled `Editing`. The main code editor area contains a single code cell with the text `1 print("Hello World!")`. To the left of the code cell is a play button icon. A yellow arrow points from the left margin to the play button. Below the code cell, the output is displayed as `Hello World!` with a green checkmark. The left sidebar shows a search icon, a variable `{x}`, and a folder icon.

https://colab.research.google.com/

chapter1_example1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

```
1 print("Hello World!")
```

Hello World!

Activity: Type the Python command as shown in the picture and observe the results.

The screenshot shows a Google Colab notebook titled "chapter1_example1.ipynb". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", along with the status "All changes saved".

The first code cell, labeled "[1]", contains the Python command:

```
1 print("Hello World!")
```

The output of this cell is "Hello World!".

The second code cell contains the following Python code:

```
1 a = "Python"  
2 b = "course"  
3 c = "is awesome"  
4 print(a,b,c)
```

Annotations on the image:

- A yellow circle with the text "2 RUN" and an arrow pointing to the play button icon on the left of the second code cell.
- A yellow circle with the text "1 CODE" and an arrow pointing to the code text in the second code cell.
- A green key icon is located on the right side of the image.

At the bottom left, the text "Co Developing Applications for Education. Asst.Prof.Dr.Nutthapat Kaewrattanapat" is visible.

Post-test

1. ภาษา Python เป็นภาษาคอมพิวเตอร์ลักษณะใด?

A

Interpreter

B

Compiler

2. ข้อใดไม่เป็นองค์ประกอบของระบบคอมพิวเตอร์

A

Hardware

D

Software

B

Peopleware

E

Data/Information

C

Communication

F

Procedures

3. "110100101010101" เทียบเคียงได้กับภาษาคอมพิวเตอร์ระดับใด?

A

Low-Level Language

B

High-Level Language

C

Assembly Language

D

Procedural Language

4. `print("Panda")` เทียบเคียงได้กับภาษาคอมพิวเตอร์ระดับใด?

A

Low-Level Language

B

High-Level Language

C

Assembly Language

D

Procedural Language

5. ข้อใด ไม่ใช่ คุณสมบัติของอัลกอริทึม

A

ไม่คลุมเครือ (Unambiguous)

B

มีขั้นตอนตายตัว (Deterministic)

C

มีจุดสิ้นสุดการทำงาน (Finiteness)

D

มีหน่วยความจำที่เพียงพอ (Memory Capacity)

6. ข้อใดไม่ใช่เครื่องมือในการโปรแกรมภาษา Python

A

Pycharm

B

Colab

C

Jupyter

D

Assembler

7. ให้ผลลัพธ์ถูกต้องตามเงื่อนไขที่กำหนดไว้เสมอ คือ คุณสมบัติข้อใด

A

ไม่คลุมเครือ (Unambiguous)

B

มีขั้นตอนตายตัว (Deterministic)

C

มีจุดสิ้นสุดการทำงาน (Finiteness)

D

มีหน่วยความจำที่เพียงพอ (Memory Capacity)

8. วิธีการสอนใดเหมาะกับการเรียนรู้ด้านการโปรแกรมคอมพิวเตอร์

A

Passivity-based Learning

B

Cognitivism

C

Experiential Learning

D

Adaptive Learning

9. การสอน Coding สิ่งแรกที่ต้องสอนให้ผู้เรียนคำนึงถึงในการโปรแกรม คืออะไร

A

Input

B

Process

C

Output

D

Memory

10. การสอน Coding สิ่งสุดท้ายที่ต้องสอนให้ผู้เรียนคำนึงถึงในการโปรแกรมคืออะไร

A

Input

B

Process

C

Output

D

Memory