

Introduction to Programming

01418112: Fundamentals Programming Concept

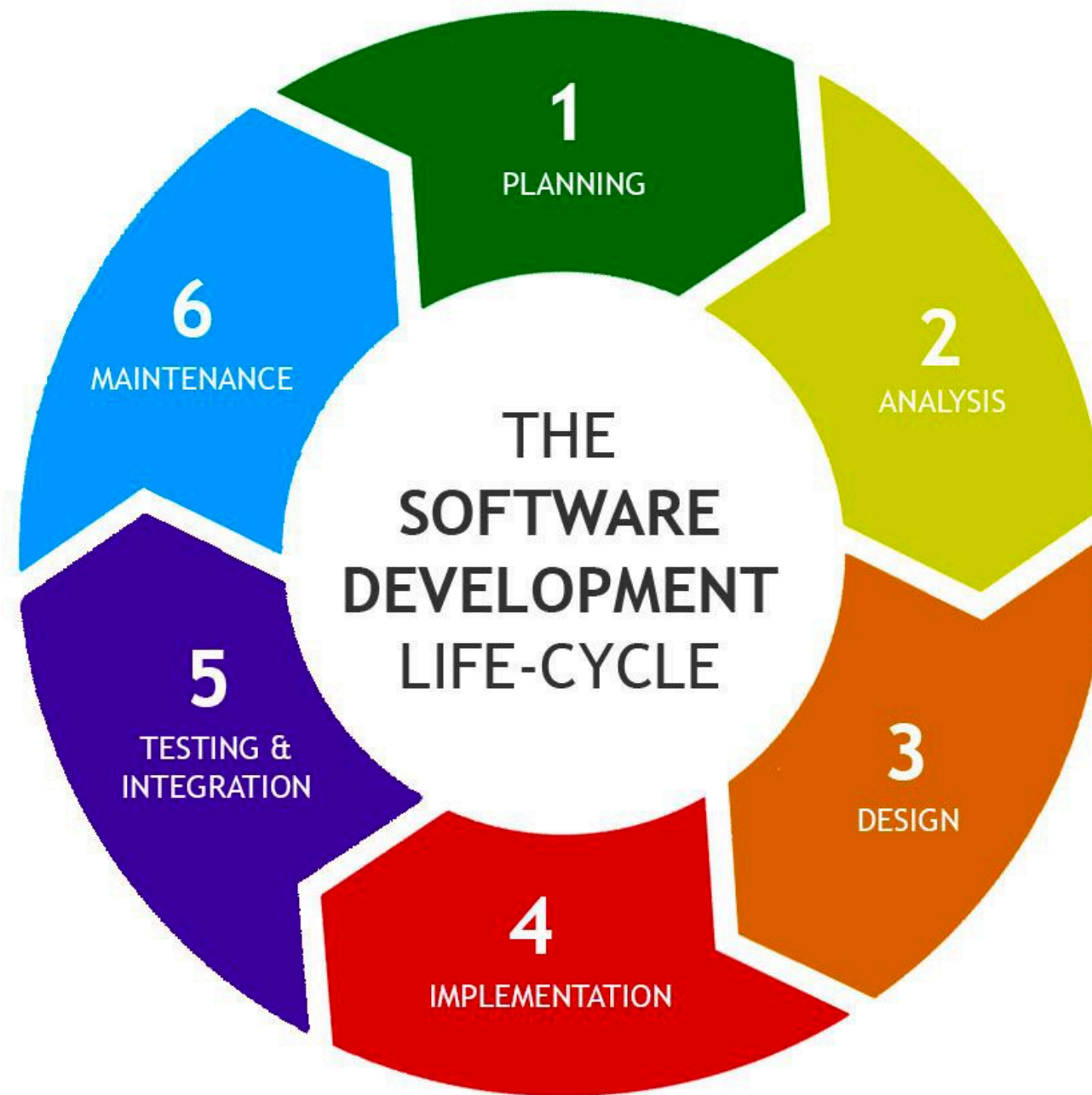
Agenda

- What is programming?
- A simple/sample program
- Prompt & input
- Variable & constant
- Assignment statement
- Arithmetic operators
- Output
- Types of data
 - Integers
 - Floating point (Real) numbers
 - Strings
- Structured Programming and flowcharts

You're already a Programmer!

- เราทุกคนฝึก “เขียนโปรแกรม” อยู่ตลอดเวลา
- ถ้ามองว่า ถ้าต้องการ smart phone เครื่องใหม่จะทำอย่างไร
 - คิดออกมาเป็นลำดับขั้นตอนของการให้ได้มา
 - ในแต่ละขั้นตอนจะทำอย่างไร
 - ลงมือทำ
- ถ้ามองว่า ถ้าต้องการแก้ไขภัยคณิตศาสตร์ซักข้อ เช่น หาค่า log จะทำอย่างไร
 - หาค่า log ของอะไร
 - คำนวณค่า log (อาจจะต้อง บวกลบคูณหาร ก่อน)
 - ได้คำตอบ
- นี่คือการโปรแกรม คอมพิวเตอร์เพียงแต่ทำงานตามคำสั่ง
- หน้าที่ของ Programmer คือ เขียนลำดับการแก้ปัญหาให้คอมพิวเตอร์ทำตามได้อย่างถูกต้อง

What is Programming?



https://miro.medium.com/max/3806/1*GU-YWDqM-ZljYHGs1iDNPg.jpeg

A Simple Sample Program

- input, constants and variables
- processing, output and types of data
- a sample problem
 - We plan to buy songs from Spotify
 - Price (in USD) is \$0.69 per song
 - We need to do price conversion
 - USD \$1 = 35.60 Baht (THB)

Price Conversion Program

- ขั้นตอนการแปลงราคาสินค้าจาก USD to THB
 1. แสดงข้อความ "Input the number of songs you wish to buy: "
 2. รับจำนวนเพลง เก็บไว้ในตัวแปร **songs**
 3. คำนวณราคารวมเป็น \$ เก็บไว้ในตัวแปร **usd**
 - 3.1. **usd** = **songs** * 0.69
 4. แปลงค่าเงินเป็นบาท เก็บไว้ในตัวแปร **thb**
 - 4.1. **thb** = **usd** * 35.6
 5. แสดง **thb**
- ขั้นตอนข้างต้นเรียกว่า **โปรแกรม** บางทีเรียกว่า **รหัสจำลอง** (Pseudo code) ซึ่งถ้าเขียนผิดเล็กน้อยก็เข้าใจแต่
- โปรแกรมคอมพิวเตอร์ ต้องเขียนในภาษาใดภาษาหนึ่ง โดยยึดตาม “**ไวยากรณ์**” (Syntax) ของภาษานั้น ไม่สามารถผิดได้เลย
- สัญลักษณ์ * แทนการคูณในโปรแกรม

Price Conversion Program

```
1 songs=int(input("Input the number of songs you wish to buy: "))
2 usd=0.69*songs
3 thb=usd*35.6
4 print("Total amount: ",thb)
```

- Check the result of the above program running...

```
Input the number of songs you wish to buy: 3
Total amount: 73.692
```

เครื่องมือที่ใช้

- เครื่องมือที่ใช้ในวิชานี้ สำหรับการเขียนโปรแกรม Python
 - ตัวแปลภาษาไพทอน – Python Interpreter
 - Search keyword: python download
 - โปรแกรมช่วยในการเขียนและรัน ไพทอน – Wing IDE 101
 - Search keyword: wing ide 101 download
- เว็บไซต์และทดสอบไพทอนออนไลน์ – Google Colab
 - Search keyword: colab

Prompt & Input

- Input prompt หรือ Prompt
 - ข้อความที่โปรแกรมแสดงให้ผู้ใช้อ่าน ก่อนจะรอรับข้อมูล
 - ไม่จำเป็นสำหรับการทำงานของโปรแกรม

```
1 songs=int(input("Input the number of songs you wish to buy: "))
2 usd=0.69*songs
3 thb=usd*35.6
4 print("Total amount: ",thb)
```

Prompt & Input (2)

- Data input
 - ข้อมูลที่เราป้อนให้กับโปรแกรม
 - ทำไมต้องมี input
 - เพราะเราไม่ทราบค่านี้ในตอนที่เราเขียนโปรแกรม
 - ต้องถูกเก็บไว้ใน ตัวแปร (variable) เพื่อให้อ้างถึงหรือนำมาใช้ได้ต่อไป
- ผู้ใช้ป้อนข้อมูลเข้าจาก keyboard ซึ่งถือเป็นอุปกรณ์อินพุตมาตรฐาน (standard input)

```
1 songs=int(input("Input the number of songs you wish to buy: "))
2 usd=0.69*songs
3 thb=usd*35.6
4 print("Total amount: ",thb)
```

Variable & Constant

- Variable หรือ ตัวแปร
 - เป็น “ชื่อ” (name) ที่กำหนดขึ้นเพื่อเก็บ “ค่า” (value)
 - ค่าที่เก็บ สามารถเปลี่ยนแปลงได้ระหว่างการทำงานของโปรแกรม
- Constant หรือ ค่าคงที่
 - ค่าไม่เปลี่ยนแปลงเลยตลอดการทำงานของโปรแกรม
 - ชนิดของค่าคงที่
 - Literal constant
 - ค่าคงที่ที่กำหนดในโปรแกรมโดยตรง ไม่มีชื่อเรียก
 - เช่น 0.69 , 35.6 , "Total amount: "
 - Name constant
 - กำหนด “ชื่อ” ให้กับค่าคงที่ ทำให้เรียกใช้โดยเรียก “ชื่อ” แทนได้
 - **NOTE:** Python ไม่มี Name constant

Variables and Constants Names

- ควรตั้งชื่อให้สอดคล้อง และมีความหมายที่แสดงถึงค่าที่ต้องการเก็บ
- จำนวนตัวอักษรไม่ควรยาวเกินที่แต่ละภาษากำหนด (ปกติยาวได้มาก)
- ไม่ใช้ตัวอักขระพิเศษ ยกเว้น underscore character และไม่ใช่ช่องว่าง
- สำหรับตัวแปร
 - ขึ้นต้นตัวแรกของชื่อตัวแปรด้วยตัวอักษรตัวเล็กเสมอ
 - ถ้าชื่อประกอบด้วยหลายคำ (หรือหลายพยางค์) ให้ขึ้นต้นคำถัดไปด้วยตัวอักษรตัวใหญ่ (หรือใช้ underscore คั่นระหว่างคำ/พยางค์) เพื่อให้อ่านได้ง่าย
- สำหรับค่าคงที่
 - ใช้ตัวอักษรตัวใหญ่ทั้งหมด
- ตัวอย่าง
 - `kmTraveled, km_Traveled, salaryPerMonth, source_1, PI, THB_PER_USD`
 - `km Traveled, 1_source`

การเก็บตัวแปรในคอมพิวเตอร์

- ตัวแปร คือ ชื่อที่อ้างถึงตำแหน่งในหน่วยความจำ

78	
Songs	DollarPrice

78	77.22
Songs	DollarPrice

Assignment Statement

- คำสั่งกำหนดค่า
 - โดยทั่วไป ในโปรแกรมคอมพิวเตอร์ จะใช้เครื่องหมาย = เป็นการกำหนดค่าให้กับตัวแปร
 - เช่น $usd = 0.69 * songs$
 - มีความหมายเป็น การกำหนดให้ค่าของตัวแปรทางซ้าย เท่ากับ ผลลัพธ์ของนิพจน์ (expression) ทางขวา ของเครื่องหมาย =
 - ทางซ้ายของเครื่องหมายกำหนดค่า ต้องเป็น ตัวแปร เท่านั้น
 - ทางขวาของเครื่องหมายกำหนดค่า ต้องเป็น นิพจน์ที่ประเมินค่าได้
 - นั่นคือ ตัวแปร นิพจน์ทางคณิตศาสตร์ ค่าคงที่ เป็นต้น
 - **ข้อควรระวัง** เครื่องหมาย = ที่ใช้ในคำสั่งกำหนดค่านี้ ไม่ได้มีความหมายเช่นเดียวกับ สมการ (equation) ในวิชาคณิตศาสตร์

Reassigning Values to Variables

- จะเกิดอะไรขึ้น...ถ้า กำหนดค่าให้กับตัวแปรที่มีค่าอยู่แล้ว เช่น

```
salaryPerMonth = 8000
```

 หรือ

```
salaryPerMonth = 9000
```
- หลังจากบรรทัดที่ 1 salaryPerMonth จะเก็บค่า 8000
- หลังจากบรรทัดที่ 2 ค่าเดิมของ salaryPerMonth จะถูกเขียนทับด้วยค่า 9000 และจะไม่เปลี่ยนแปลงไปจนกว่าจะถูกเปลี่ยนอีกครั้งหนึ่ง
- กำหนดค่าของตัวแปรหนึ่ง ให้กับตัวมันเองก็ได้ เช่น
 - ```
count = count + 1
```
  - ถ้าก่อนหน้านี้คำสั่งนี้ count เก็บค่า 28  
หลังจากคำสั่งนี้ count จะเก็บค่า 29



# Arithmetic Operators

**Table 1.1** Arithmetic Operators

| Operator       | Computer Symbol | Example       |
|----------------|-----------------|---------------|
| Addition       | +               | $2 + 3 = 5$   |
| Subtraction    | -               | $7 - 3 = 4$   |
| Multiplication | *               | $5 * 4 = 20$  |
| Division       | /               | $12 / 3 = 4$  |
| Exponents      | **              | $2 ** 3 = 8$  |
| Modulus        | %               | $14 \% 4 = 2$ |

$$C = \frac{5(F - 32)}{9}$$

$$C = 5 * (F - 32) / 9$$



# Operator Precedence

- ลำดับความสำคัญของ arithmetic operator
  - วงเล็บ       $**$        $*/\%$        $+ -$
- หาค่าของนิพจน์ต่อไปนี้เปรียบเทียบกัน
  - $6 + 8 / 2 * 4$
  - $6 + 8 / (2 * 4)$
  - $(6 + 8) / 2 * 4$
  - $(6 + 8) / (2 * 4)$

# การแสดงผลลัพธ์

- ตัวอย่างคำสั่งใน pseudo code เพื่อแสดงผลลัพธ์
  - แสดง **thb**
- ตัวอย่างคำสั่ง Python เพื่อแสดงผลลัพธ์
  - `print("Total amount: ", thb)`
- เป็นการแสดงผลลัพธ์ คือ ค่าของตัวแปรที่ระบุออกทางจอภาพ ซึ่งเป็นอุปกรณ์เอาต์พุตมาตรฐาน (standard output)
- ผลลัพธ์จะถูกพิมพ์ออกมา เริ่มต้นจากตำแหน่งที่ cursor อยู่

# Output Annotation

- การอธิบายผลลัพธ์
- ที่ผ่านมา ผลลัพธ์จะแสดงเฉพาะค่าของตัวแปรเท่านั้น
- ผู้พัฒนาโปรแกรม อาจจะเลือกแสดงข้อความอธิบายความหมาย
  - เช่น ถ้าต้องการให้แสดงผลลัพธ์
    - Total amount in Thai Baht is 73.89 baht.
  - ภาษา Python
    - `print("Total amount in Thai Baht is %.2f baht." % thb)`
- การแสดง หรือไม่แสดงค่าอธิบายผลลัพธ์ ขึ้นอยู่กับคำสั่งของโจทย์ หรือความต้องการของผู้ใช้โปรแกรม

# Types of Data

- การใช้ตัวแปรต้องระบุชนิดของข้อมูลที่ต้องการเก็บ
  - ข้อมูลแต่ละชนิดเก็บในคอมพิวเตอร์เป็นเลขฐานสอง
    - นั่นคือข้อมูลแต่ละประเภท อยู่ในรูปแบบเป็น bit pattern
  - ข้อมูลแต่ละชนิดต้องการพื้นที่เก็บไม่เท่ากัน
- Numeric data
  - Integers
  - Real or Floating point numbers
- Text
  - Characters
  - Strings

# Integers

- เก็บค่าได้เฉพาะจำนวนเต็มเท่านั้น
- ถูกเก็บในหน่วยความจำในรูปแบบ เลขจำนวนเต็มฐานสอง (bit pattern)
- การบวก ลบ คูณ เลขจำนวนเต็ม ผลลัพธ์จะเป็นจำนวนเต็ม
- แต่ ผลการหารเลขจำนวนเต็มสองจำนวน อาจจะได้จำนวนเต็ม เช่น
- ใน Python (version 3+) จะได้เป็นจำนวนจริง (floating point number)
  - ตัวอย่าง ถ้า  $a=8$  และ  $b=3$   
 $a/b$  จะได้ค่าเป็น 2.6666666666666665

# Floating Point Numbers

- เก็บค่า “โดยประมาณ” ของเลขที่มีทศนิยม
- Numbers with fractional parts
  - 8.0, 3.1415926, 4.3 x 10<sup>3</sup>
- การเลือกใช้งาน integers หรือ floating point numbers
  - ขึ้นอยู่กับคุณสมบัติของข้อมูลที่จัดเก็บ
- Python
  - ไม่จำเป็นต้องประกาศชื่อตัวแปรก่อน
  - ไม่ต้องกำหนดชนิด ตัวแปรสามารถใช้อ้างถึงค่าที่ต้องการได้เลย

# Strings

- **สายอักขระ** ประกอบด้วย ตัวอักขระ (character) ตั้งแต่ 0 ตัวขึ้นไปมาเรียงต่อกัน
- โดยทั่วไป ในภาษาโปรแกรมต่าง ๆ กำหนดให้เขียน Literal constant ของ string อยู่ในเครื่องหมาย " " เช่น
  - "Computer", "123", " ", ""
  - Python สามารถใช้เครื่องหมาย ' ', " ", หรือ ''' '''
- Null string
  - สายอักขระว่าง มีตัวอักขระจำนวน 0 ตัว
- String concatenation (การเชื่อม string)
  - วิธีการแตกต่างกันไปในแต่ละภาษา
  - ใน Python ใช้เครื่องหมาย + เพื่อเชื่อมสตริง
  - "Computer" + "123"




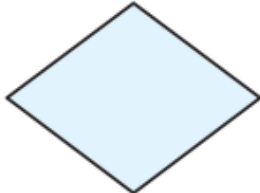

# Structured Programming

- Tools to describe algorithms
  - เครื่องมือในการอธิบาย อัลกอริทึม (ขั้นตอนวิธีในการแก้ปัญหา)
- Natural Language – ใช้ภาษาทั่วไป
  - ง่าย แต่เยิ่นเย้อและทำให้สับสนได้ง่าย
- High Level Computer Programming Language
  - เป็นขั้นตอน รัดกุม แต่ยากต่อการเขียนสำหรับผู้เริ่มต้น และทำให้ต้องแบ่งแยกสมาธิมากกังวลกับเรื่องไวยากรณ์ของภาษา ทำให้ไม่สามารถสนใจกับปัญหาได้เต็มที่
- Pseudo code – รหัสเทียม รหัสจำลอง
  - ไม่มีข้อกำหนดที่ตายตัว ทำให้สามารถให้ความสนใจกับปัญหาที่กำลังคิดแก้ไขได้เต็มที่ และแปลงไปเป็นภาษาโปรแกรมหรือ flow chart ได้ง่าย
- Flowchart
  - มีข้อกำหนดและรูปแบบแน่นอน แต่ไม่ซับซ้อนเท่ากับภาษาโปรแกรม แปลงไปเป็นภาษาโปรแกรมได้ง่ายกว่า pseudo code



# Flowcharts

**Figure 2.5** Basic flowcharting symbols

| Symbol                                                                              | Name         | Description                                                            |
|-------------------------------------------------------------------------------------|--------------|------------------------------------------------------------------------|
|    | Terminator   | Represents the start or end of a program or module                     |
|   | Process      | Represents any kind of processing function; for example, a computation |
|  | Input/output | Represents an input or output operation                                |
|  | Decision     | Represents a program branch point                                      |
|  | Connector    | Indicates an entry to, or exit from, a program segment                 |

# Control Structures

- โครงสร้างการควบคุม
  - โครงสร้างการทำงานพื้นฐานของโปรแกรม
  - จำนวนน้อยที่สุดที่เพียงพอต่อการออกแบบอัลกอริทึมเพื่อแก้ปัญหา
    1. Sequential structure
    2. Decision (selection) structure
    3. Loop (repetition) structure

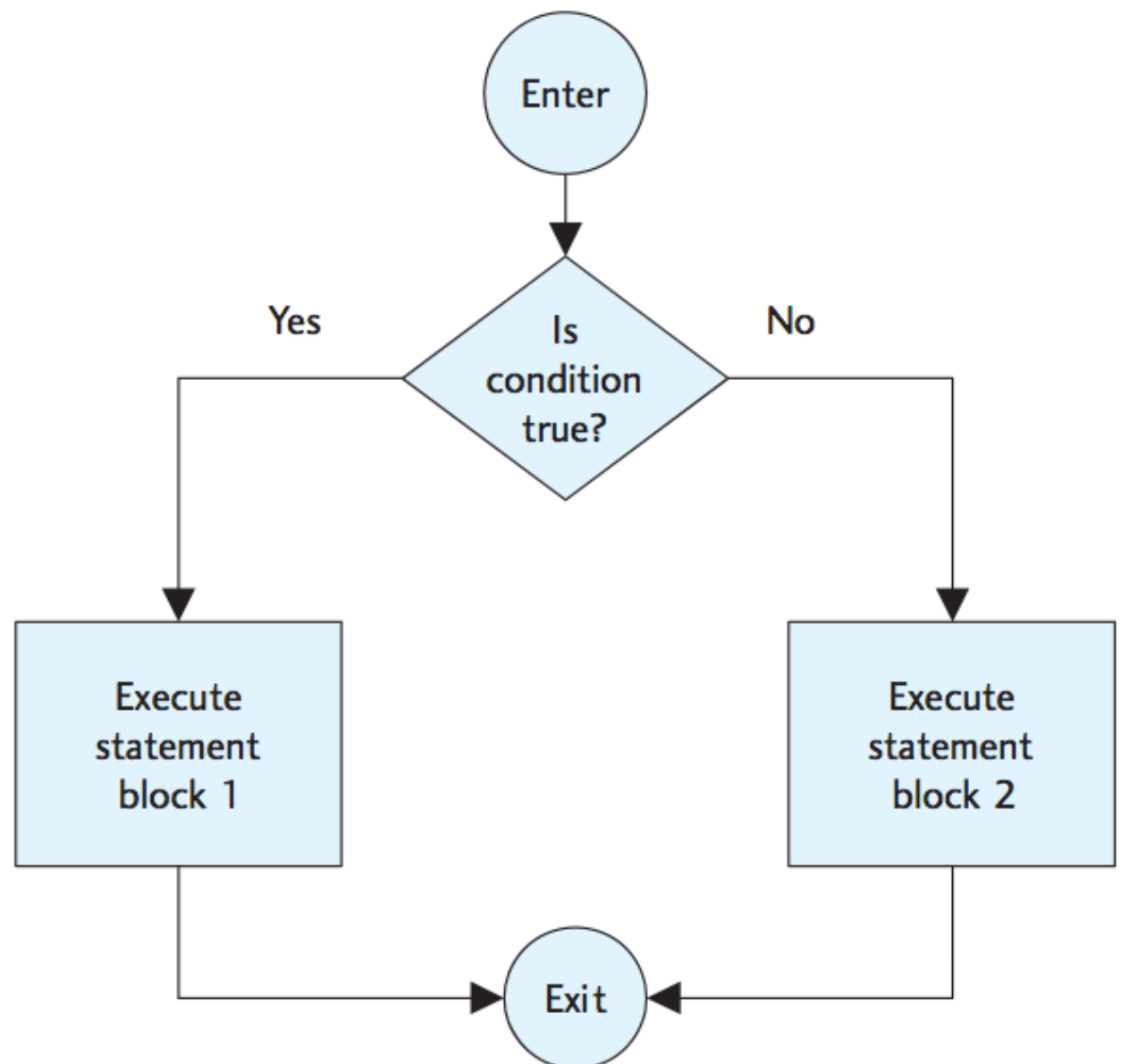
# Sequential Structures

- Sequential structures
  - ประกอบด้วยคำสั่งต่าง ๆ เรียงต่อเนื่องกัน
  - เป็นการทำงานพื้นฐานที่สุดของ structured programming
- กระบวนการทำงานพื้นฐาน 3 ชนิด
  - Computation
    - การคำนวณ รวมถึงการกำหนดค่าให้กับตัวแปร เพื่อให้สามารถอ้างอิงและนำมาใช้ในภายหลังได้ เช่น assignment statement
  - Input
    - การรับข้อมูลเข้า จากอุปกรณ์รับข้อมูลเข้า เช่น แป้นพิมพ์ (keyboard) เพื่อนำไปให้ค่ากับตัวแปร และเก็บไว้ในหน่วยความจำ
  - Output
    - การส่งข้อมูลออก เพื่อไปแสดงผลยังอุปกรณ์แสดงผล เช่น จอภาพหรือเครื่องพิมพ์ ข้อมูลที่จะส่งออกโดยทั่วไปจะเป็นค่าคงที่ หรือค่าตัวแปรซึ่งอ่านมาจากหน่วยความจำนั่นเอง

# Decision Structures

- Decision structures
  - ต้องมีการประเมินเงื่อนไข แล้วเลือกชุดคำสั่งในการทำงานตามผลของการประเมินที่ได้

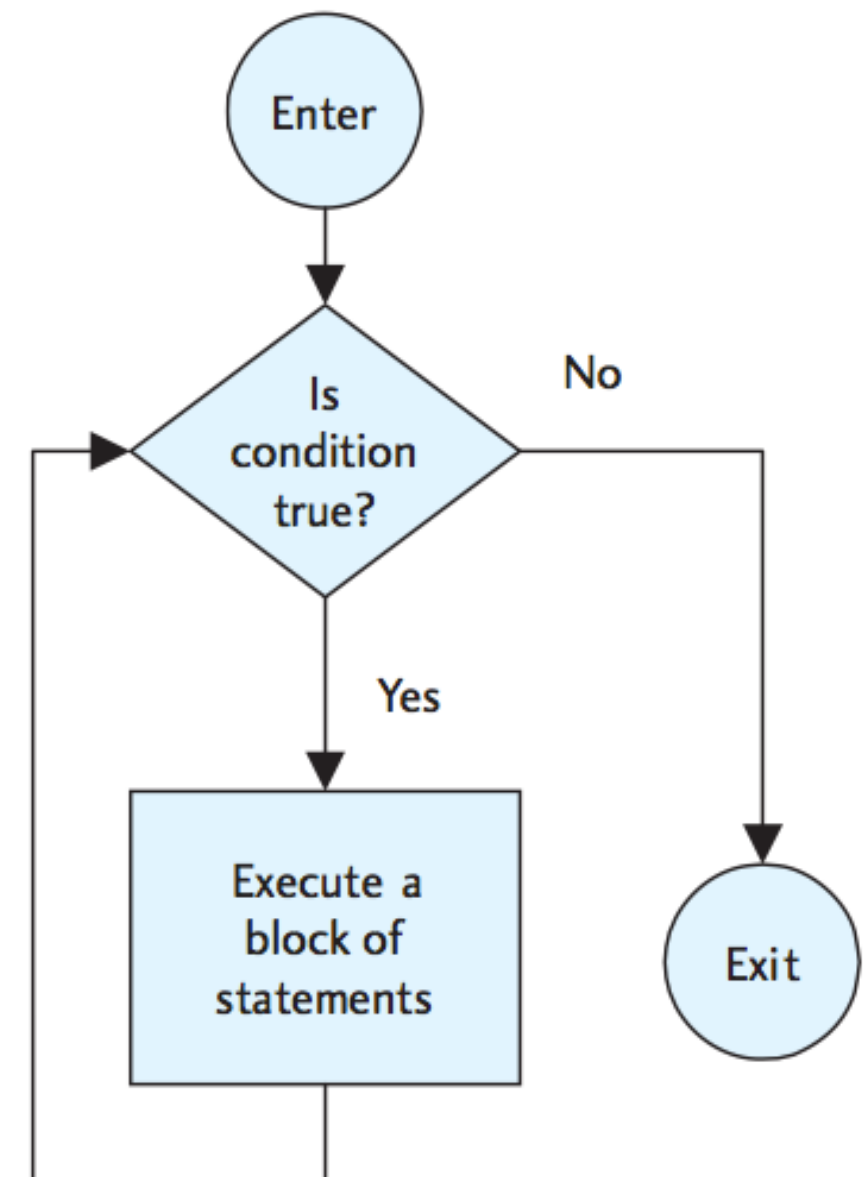
**Figure 2.8** Flowchart for a typical decision structure



# Loop Structures

- Loop structures
  - ต้องมีการประเมินเงื่อนไข แล้วย้อนกลับไปทำคำสั่งเดิมซ้ำอีกจนกว่าผลของเงื่อนไขจะเปลี่ยนแปลง

**Figure 2.7** Flowchart for a typical loop structure



# Flowchart for Three Constructs

