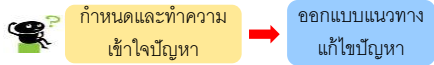


ประเภทของปัญหา- BASIC CONCEPT



• สิ่งที่ต้องทำความเข้าใจเมื่อต้องการแก้ปัญหา:

- กำหนดปัญหา
- เข้าใจปัญหา
 - Input
 - ผลลัพธ์ที่ต้องการ
- ออกแบบอัลกอริธึม (Algorithm)
 - การกระทำ
 - ลำดับของแต่ละการกระทำ
 - การกระทำทั้งหมดต้องมีวันสิ้นสุด



Process of Problem Solving

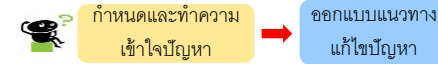
Dept. of Computer Sciences, Thammasat University

1

LAST LECTURE – SOLVING PROBLEM

Example Problem:

การคำนวณผลรวมของเลขที่รับเข้ามา 2 จำนวน

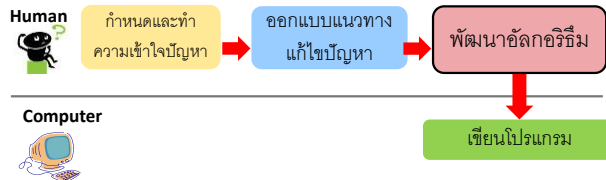


- Input: รับเลข 2 จำนวน (กำหนดให้เป็นตัวแปร x และ y)
- Expected Output: ผลรวมของ x และ y
- Action: 1. คำนวณผลรวมโดยเก็บคำตอบในตัวแปร result = x+y
2. แสดงผลรวมใน result (optional)



2

THIS LECTURE



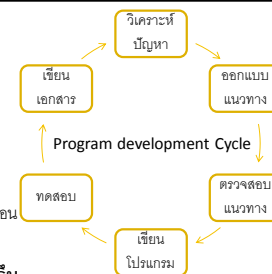
- วัฏจักรการพัฒนาโปรแกรม
- เครื่องมือที่ช่วยในการพัฒนาโปรแกรม
 - Pseudo-code
 - Flowchart
- ออกแบบโครงสร้างการควบคุมโดยใช้ Flowchart

3

วัฏจักรการพัฒนาโปรแกรม 1

วิเคราะห์ปัญหา คำตอบ และข้อจำกัดต่าง ๆ

- เข้าใจปัญหาอย่างถ่องแท้
- เข้าใจข้อจำกัดต่าง ๆ ของปัญหา
- แบ่งปัญหาเป็นปัญหาย่อย ๆ หากปัญหามีความซับซ้อน



ออกแบบและตรวจสอบความถูกต้องของอัลกอริธึม

- ออกแบบแนวทางการแก้ไขปัญหาลำหรับปัญหาย่อย
- ตรวจสอบความถูกต้องของอัลกอริธึม (แนวทางการแก้ไขปัญหาย่อย) ที่ออกแบบไว้
 - ใช้ข้อมูลตัวอย่าง
 - ตรวจสอบความถูกต้องของสูตรที่ใช้ในการคำนวณ

4

วัฏจักรการพัฒนาโปรแกรม 2

เขียนโปรแกรมตามที่ออกแบบไว้

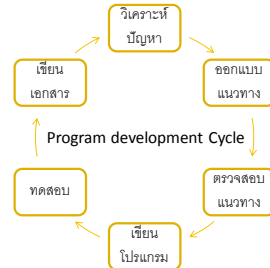
- ตรวจสอบความถูกต้องของอัลกอริทึมแล้ว

ทดสอบโปรแกรม

- ใช้หลาย ๆ กรณีสำหรับทดสอบ

เขียนเอกสาร

- เขียนคอมเมนต์ในโค้ด



5

การวิเคราะห์และออกแบบแนวทางการแก้ไข้ปัญหา

วิเคราะห์ปัญหา คำตอบ และข้อจำกัดต่าง ๆ

Input	Processing	Output
Regular Time Hours Worked	Read regular time hours worked, overtime hours worked, hourly pay rate.	Gross Pay
Overtime Hours Worked	Calculate regular time pay.	
Hourly Pay Rate	If employee worked overtime, calculate overtime pay.	
	Calculate gross pay.	
	Print gross pay.	

- Ex. คำนวณเงินเดือนของพนักงาน

- Input Process Output (IPO) chart

- IPO chart – ใช้ระบุ inputs, outputs, and processing steps

ออกแบบแนวทางการแก้ไข้ปัญหา: ออกแบบอัลกอริทึม

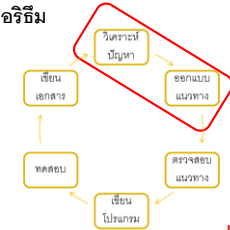
- Hierarchy chart

- แสดงแต่ละส่วนของโปรแกรมโดยใช้แผนภาพ

- เรียกอีกอย่างว่า structure chart

- Flowchart

- Pseudocode

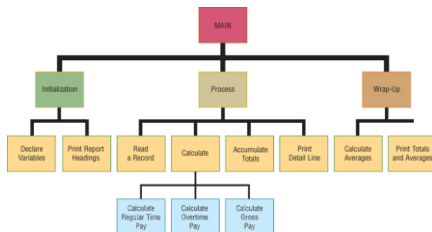


6

ออกแบบแนวทางการแก้ไข้ปัญหา : HEIRARCHY CHART

แนวทางแก้ปัญหาอย่างมีประสิทธิภาพ โดยใช้คอมพิวเตอร์

1. แดกปัญหาใหญ่เป็นส่วนย่อย เรียกว่า โมดูล (modules)
2. งานใดที่ต้องทำบ่อย ให้สร้างเป็นโมดูล



7

ออกแบบแนวทางการแก้ไข้ปัญหา : PSEUDO-CODE

- Pseudo-Code:

- ใช้ภาษาที่มนุษย์เข้าใจในการพัฒนาอัลกอริทึม
- ไม่ใช่ภาษาที่ใช้ในการเขียนโปรแกรม
- ช่วยให้โปรแกรมเมอร์ออกแบบโปรแกรมก่อนพยายามเขียนโปรแกรมจริง

Example of keywords:

รับ ใช้เมื่อต้องการรับข้อมูลจากผู้ใช้

แสดง ใช้เมื่อต้องการแสดงข้อความแก่ผู้ใช้

คำนวณ ใช้เมื่อต้องการคำนวณ โดยสามารถใช้สัญลักษณ์ทางคณิตศาสตร์

เช่น + - * / ()

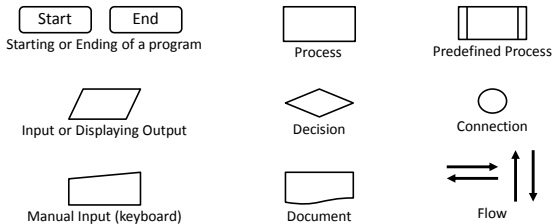
เก็บค่า เก็บค่าใส่ตัวแปรสำหรับใช้ภายหลัง

8

ออกแบบแนวทางการแก้ไขปัญา : FLOWCHART

Flowchart:

- ใช้เพื่อนำเสนออัลกอริทึมโดยใช้แผนภาพ
- ใช้สัญลักษณ์เฉพาะเพื่อแสดงส่วนของโปรแกรม
- ข้อดี: สามารถแสดงลำดับการทำงานได้ชัดเจน



Examples of special-purpose symbols used for representing an algorithm

9

PSEUDO-CODE AND FLOWCHART

Input : ค่า 2 ค่า (จากผู้ใช้)

กำหนดให้เก็บค่าแรกในตัวแปร x และ เก็บค่าที่ 2 ในตัวแปร y

Output : แสดงค่าตอบของการคำนวณผลรวมของค่า x และ y

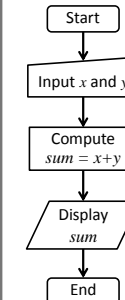
กำหนดให้ตัวแปร sum ใช้เก็บผลการคำนวณ

Action : คำนวณผลรวม $sum = x + y$

Problem Analysis of A summation of Two Values



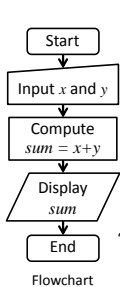
Pseudo-Code



Flowchart

10

ALGORITHM DEVELOPMENT



Flowchart

รับ ค่า x จากผู้ใช้
รับ ค่า y จากผู้ใช้
คำนวณ ค่า $sum = x + y$
แสดง ค่า sum

Pseudo-Code

- ตรวจสอบ แนวทางการแก้ไขปัญาที่ออกแบบไว้สามารถแก้ไขทุกกรณีหรือไม่

จาก **pseudo-code** หรือ **flowchart** ได้คิดโปรแกรมภาษา C มีลำดับ ดังนี้:

- เตรียมตัวแปร 3 ตัว โดยตั้งชื่อว่า x , y , และ sum
- กำหนดค่าให้กับตัวแปร x และ y จากค่าที่ผู้ใช้กำหนดทางคีย์บอร์ด
- คำนวณค่าให้กับตัวแปร sum โดยคำนวณ $x + y$
- แสดงค่าที่เก็บไว้ในตัวแปร sum ผ่านทางหน้าจอ

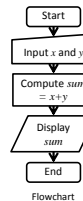
จบโปรแกรม



Problem Solving Process

11

ALGORITHM DEVELOPMENT – SOURCE CODE



Flowchart

From the pseudo-code or the flowchart, the C program must:

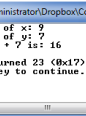
- เตรียมตัวแปร 3 ตัว โดยตั้งชื่อว่า x , y , และ sum
- กำหนดค่าให้กับตัวแปร x และ y จากค่าที่ผู้ใช้กำหนดทางคีย์บอร์ด
- คำนวณค่าให้กับตัวแปร sum โดยคำนวณ $x + y$
- แสดงค่าที่เก็บไว้ในตัวแปร sum ผ่านทางหน้าจอ



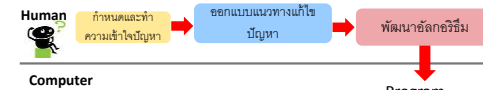
Human



Computer



Output of the program



```

1 #include <stdio.h>
2
3 main()
4 {
5     int x, y, sum;
6
7     printf("Input value of x: ");
8     scanf("%d", &x);
9
10    printf("Input value of y: ");
11    scanf("%d", &y);
12
13    sum = x + y;
14    printf("Result of %d + %d is: %d\n", x, y, sum);
15
16    return;
17 }
18

```

12

EXAMPLE 1

Problem: พัฒนาโปรแกรมสำหรับแปลงอุณหภูมิ Celsius ที่รับจากผู้ให้เป็น Fahrenheit ดังตัวอย่าง

30 Celsius คือ $(30 * 9/5) + 32$ Fahrenheit

Problem Analysis:

Input:

Expected Output:

Actions:

Pseudo-code or a Flowchart:



EXAMPLE 2

Problem: เขียนโปรแกรมที่รับค่าจากผู้ให้ จากนั้นเปรียบเทียบค่า หากค่าที่รับเข้ามามีค่าน้อยกว่า 0 ให้แสดงคำว่า **Negative** ทางหน้าจอ มิฉะนั้นให้แสดงคำว่า **Positive**

Problem Analysis:

Input:

Expected Output:

Actions:

Pseudo-code or a Flowchart:



EXAMPLE 3

Problem: เขียนโปรแกรมสำหรับคำนวณผลรวมของตัวเลข 5 ตัว โดยมีข้อกำหนดว่าสามารถคำนวณผลรวมได้ทีละ 2 ค่า เท่านั้น เช่น $c = a + b$ จากนั้นคำนวณ $c = c + d$

Example: ผลรวมของ 1, 9, 5, 10, 3 คือ 28

กำหนดให้ a_i เป็นตัวแปรสำหรับเลขแต่ละตัว โดย i มีค่าตั้งแต่ 1 ถึง 5

Problem Analysis:

Input:

Expected Output:

Actions:

Pseudo-code or a Flowchart:

จะต้องแก้ไข **Flowchart** อย่างไร เพื่อหาผลรวมของเลข 7 จำนวน ?



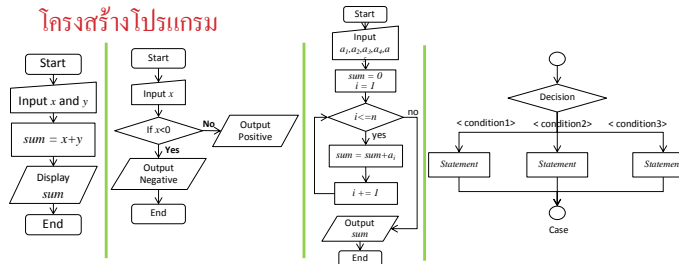
THIS LECTURE

Program development process

- Algorithm Development
 - Pseudo-code
 - Flowchart
- Control Structure from Flowchart

INTRODUCTION TO CONTROL STRUCTURES

โครงสร้างโปรแกรม

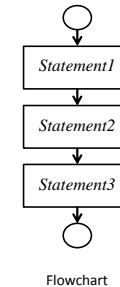


จาก flowchart ด้านบน โครงสร้างการควบคุมสามารถแบ่งได้เป็น 4 ประเภท

- **Sequential Structure** การทำงานแบบตามลำดับ
- **Decision Structure** การตัดสินใจทำตามเงื่อนไข
- **Loop Structure** การทำซ้ำ
- **Case Structure** การทำงานตามกรณี

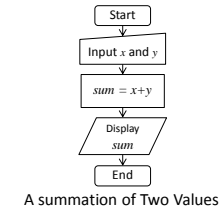
19

CONTROL STRUCTURE - SEQUENTIAL



Flowchart

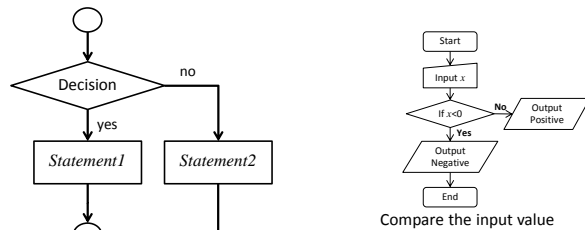
- การทำงานแบบตามลำดับ
- แต่ละคำสั่งถูกวางเรียง และ ทำงาน เป็นลำดับตามที่วางไว้



A summation of Two Values

20

CONTROL STRUCTURE - DECISION



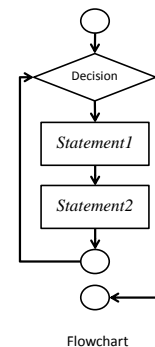
Compare the input value

Flowchart

- การตัดสินใจทำตามเงื่อนไข
- แต่ละคำสั่งจะทำงานหรือไม่ขึ้นอยู่กับเงื่อนไข
- **From the flowchart,**
 - คำสั่งที่ 1 ถูกกระทำหากเงื่อนไขเป็นจริง (true)
 - มิฉะนั้น คำสั่งที่ 2 ถูกกระทำ

21

CONTROL STRUCTURE - LOOP



Flowchart

- การทำซ้ำ
- คำสั่ง หรือ ชุดคำสั่ง กระทำซ้ำมากกว่า 1 ครั้ง
- หยุดทำคำสั่งเมื่อเงื่อนไขเป็นเท็จ

22

EXAMPLE 4

Problem: ให้ผู้ใช้ใส่จำนวนตัวเลขที่ต้องการคำนวณผลรวม กำหนดให้เป็นตัวแปร n จากนั้นให้ผู้ใช้ใส่ค่า ตามจำนวนตัวเลขที่ต้องการคำนวณผลรวม คำนวณผลรวมและแสดงผลผ่านทางหน้าจอ

Example: ผลรวม 7 จำนวน ตัวเลขที่ใส่คือ 1, 5, 10, 3, 7, 10, 2 ผลรวมคือ 38

กำหนดให้ a_i เป็นตัวแปรสำหรับเลขแต่ละตัว โดย i มีค่าตั้งแต่ 1 ถึง n

Problem Analysis:

Input:

Expected Output:

Actions:

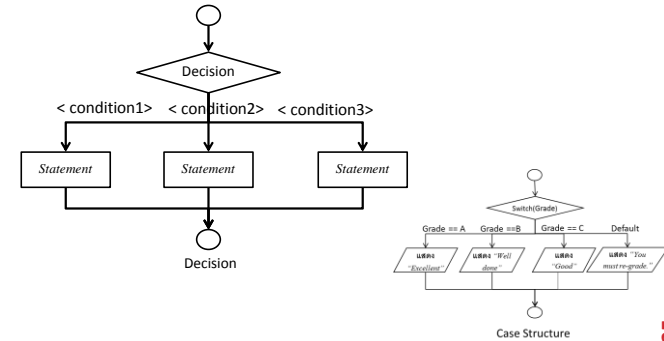
Pseudo-code or a Flowchart:



23

CASE STRUCTURE การทำงานตามกรณี

- Case Structure:** เปรียบเทียบค่าของแต่ละเงื่อนไขที่กำหนด หากเงื่อนไขเป็นจริง (true) กระทำคำสั่งนั้น ๆ



25

EXAMPLE

Problem: เขียนโปรแกรมเพื่อรับจำนวนเงินจากผู้ใช้ จากนั้นเปรียบเทียบจำนวนเงินที่รับมากและราคาของสินค้า 4 อย่าง ที่กำหนดด้านล่าง แสดงรายการสินค้าที่ผู้ใช้สามารถซื้อได้

Drinks	Price (Baht)
Coke	15
Fanta Melon	15
Water	10
Coffee Late	25



Problem Analysis:

Input: Money x , n drinks, and price p_i of a drink d_i

Expected Output:

Actions:

Pseudo-code or a Flowchart:

26

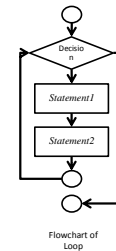
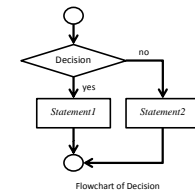
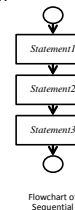
CONCLUSION

Algorithm Development

- Pseudo-code and Flowchart
- Symbol for creating flowchart

Control Structure

- Sequential
- Selection
- Loop



29