

Chapter 4:

An Introduction to Programming Structure

Somchai Cheingpongpan

Department of Information Technology
Faculty of Industrial Technology and Management

Objectives

1. Explain the need for structured programming.
2. Explain how to design modules and functions in terms of cohesion and coupling
3. Explain the difference between logical and global variables.
4. Explain the use of parameters
5. List and describe the three logic structure : sequential, decision, and loops.

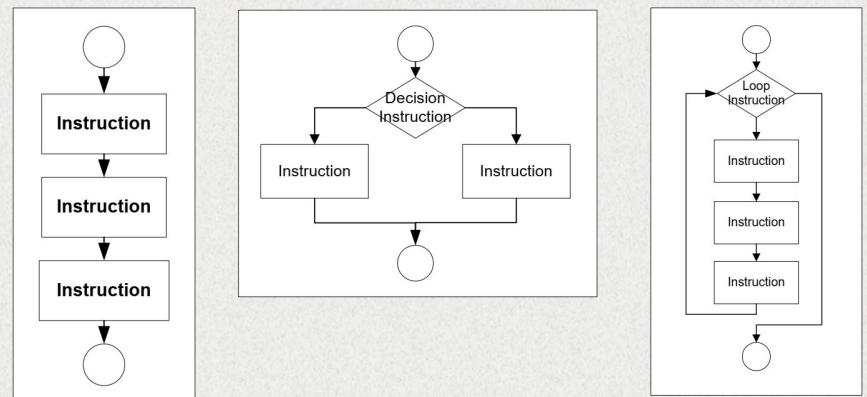
Contents

- ❑ Pointers for Structuring a Solution
- ❑ The Modules and Their Functions
- ❑ Cohesion and Coupling
- ❑ Local and Global Variables
- ❑ Parameters
- ❑ Return Values
- ❑ Variables Names and the Data Dictionary
- ❑ The Three Logic Structures

Pointers for Structuring a Solution

- ❑ Use Modules
- ❑ Use the Three Logic Structures
 - ❖ **sequential structure**
 - ❖ **decision structure**
 - ❖ **loop structure**
- ❑ Eliminate the Rewriting of Identical Process by Using Modules
- ❑ Use Techniques to Improve Readability

Pointers for Structuring a Solution(con)



The Modules and Their Functions

- เป็นขั้นตอนการแยกปัญหาออกเป็นโมดูล แต่ละโมดูล จะแทนด้วยหน้าที่การทำงานเฉพาะอย่าง
- โมดูลจะถูกประมวลผลตามลำดับที่กำหนดไว้ในผัง โครงสร้าง โดยโปรแกรมเมอร์เขียน algorithm ในแต่ละโมดูล
- มีการทดสอบกับโมดูลที่เขียนขึ้นให้ถูกต้องก่อนเขียน โมดูลถัดไป ชึ้งการเขียนและทดสอบกับโมดูลที่ขนาดเล็กจะง่ายกว่าการทำกับโมดูลขนาดใหญ่

The Modules and Their Functions(con)

กฎเกณฑ์สำหรับการออกแบบโมดูล เพื่อให้ออกแบบทาง แก้ปัญหา

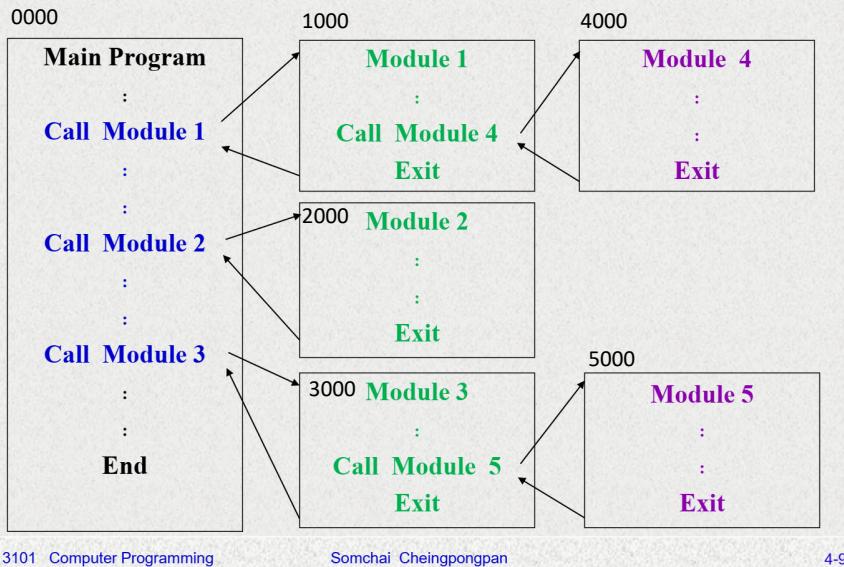
- Each module is an entity in itself
- Each module has a single function
- Each module is short enough to be easily read and modified
- The length of a module is governed by its function and the number of instructions
- A module is developed to control the order of processing

The Modules and Their Functions(con)

สามารถแบ่งประเภทโมดูลได้ ดังนี้

- **Control Module**
- **Initialization Module**
- **Process Data Module**
 - ❖ Calculation Modules
 - ❖ Print Modules
 - ❖ Read and Data Validation Modules
 - ❖ File Maintenance Modules
- **Wrap-up Module**

The Modules and Their Functions(con)



Cohesion and Coupling

□ Cohesion

เป็นการแสดงความสัมพันธ์ของหน้าที่การทำงานในแต่ละโมดูลที่เกี่ยวข้องกันของการแก้ปัญหานั้น เพื่อเป็นการแยกปัญหานั้นออกเป็นส่วนๆ ซึ่งแต่ละส่วนถูกออกแบบเป็นโมดูลเฉพาะงานที่สมบูรณ์ในตัวเอง โดยให้แต่ละโมดูลเป็นอิสระต่อกันไม่เกี่ยวข้องกับโมดูลอื่นๆ ที่มีทางเข้าหนึ่งทางและทางออกหนึ่งทาง

และสามารถทดสอบการทำงานโมดูลได้อย่างอิสระแบบ Black Box

Cohesion and Coupling(con)

□ Coupling

เป็นลักษณะที่ให้โมดูลนั้นสามารถสื่อสารระหว่างโมดูล และมีการเชื่อมต่อกันได้ โดยยอมให้โปรแกรมเมอร์สามารถส่งข้อมูลจากโมดูลหนึ่งไปยังโมดูลอื่นได้

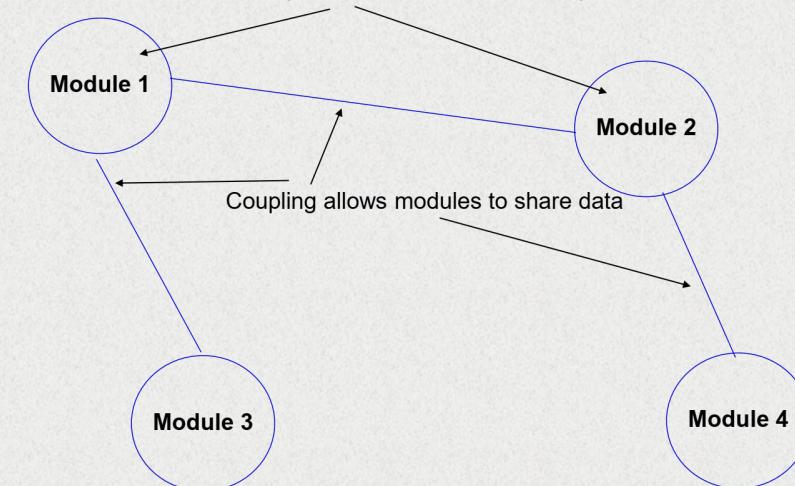
โดยเทคนิคของ coupling นี้ จะมี 3 ส่วน คือ functions, function name และ data

Cohesion และ Coupling คือเป็นหลักการที่สำคัญในการเขียนโปรแกรม

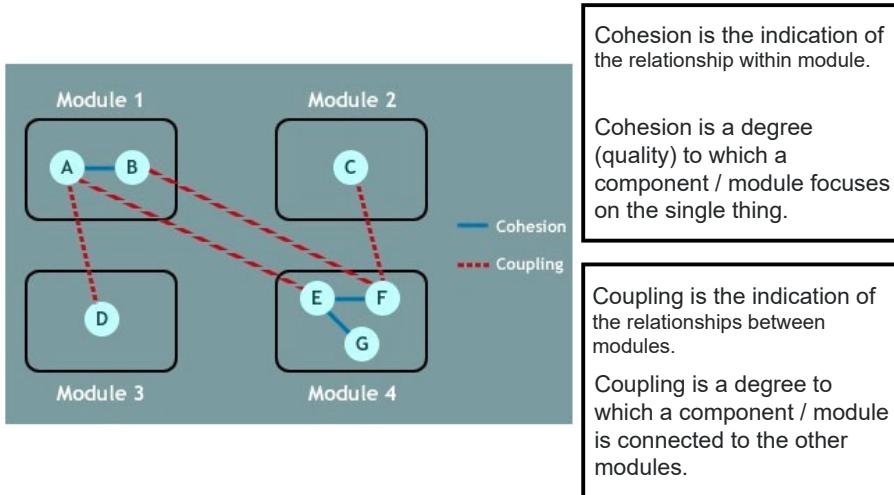
สำหรับภาษาคอมพิวเตอร์ในการจัดแบ่งงานจะเรียกว่า Dividing and Conquering

Cohesion and Coupling(con)

Cohesion is the ability for each module to be independent of other modules



Good software design has high cohesion and low coupling.



Cohesion Form

Functional Cohesion is when all of the parts of a class or module are focused on a single task. This is [the highest form of cohesion](#).

Sequential Cohesion is when modules are grouped such that each modules output feed the next modules input.

Communication Cohesion is when each module operates on the same type of data (e.g., a personnel record).

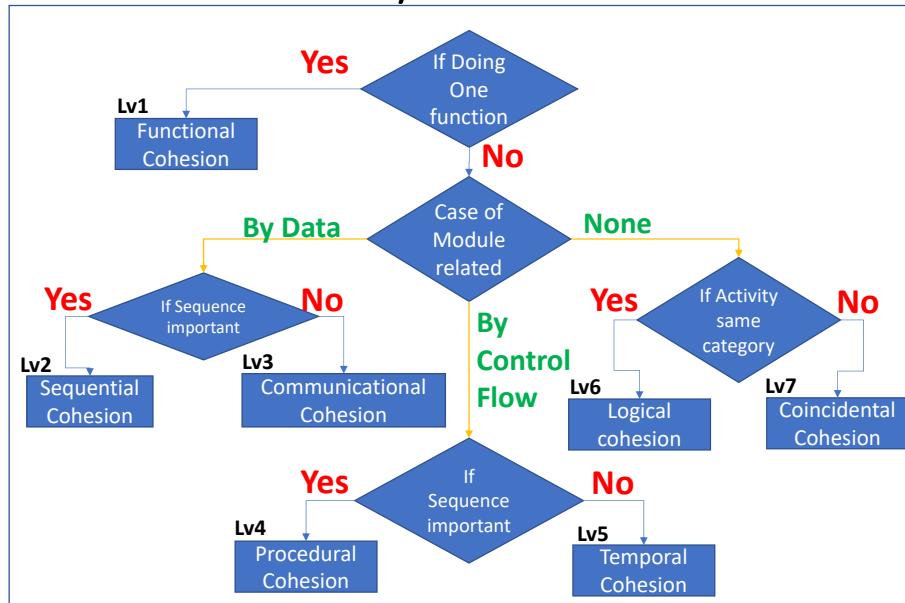
Procedural Cohesion occurs when modules are grouped together because they are all part of a sequence of operation to perform a complex procedure.

Temporal Cohesion is when modules are grouped together because they all tend to function at the same time in a program's execution.

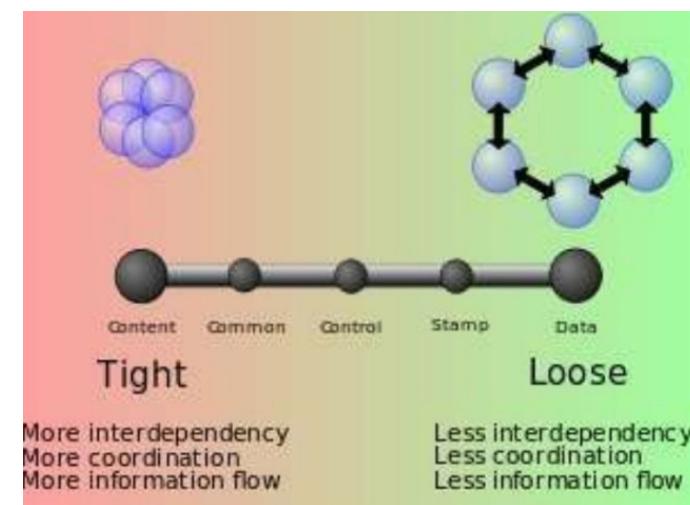
Logical Cohesion is when modules are grouped together because they perform similar functions (e.g., graphics routines).

Coincidental Cohesion is when modules are grouped randomly and [have no significant relationship](#). This is the [lowest form of cohesion](#).

How to Analysis Cohesion Level



Coupling is a [measurement of the level of dependency between modules](#). Low coupling is desired because it [increases the maintainability and readability of the module](#). There are many types of coupling (shown lowest to highest)



Coupling หมายถึง การเชื่อมต่อ นั่นหมายความว่า **การที่เกิด coupling เยอะๆ ในระบบ** ของเราจะทำให้ code ติดกันเป็นก้อน เวลาทำการย้าย หรือ แก้ไขก็ทำได้ลำบาก Coupling เป็นการบอกถึงระดับความเกี่ยวข้องกันของ Object ต่างๆ ในระบบ หากระบบใดมีความเกี่ยวข้องกันของ Object ที่สูงก็จะทำให้ระบบนั้นมีความอิสระน้อยลง

เช่น ใน Class ที่เราสร้างขึ้นมา มีการอ้างไปถึง Class อื่น ๆ อีก ถ้าเกิด Class นั้นมีการเปลี่ยนแปลงขึ้นมาก ก็จะส่งผลกระทบถึง Class ดังนั้น **ควรให้องค์ประกอบในระบบมีความเกี่ยวข้องกันน้อยที่สุด เมื่อมีการแก้ไข ส่วนใดส่วนหนึ่ง ก็จะกระทบกับส่วนอื่น ๆ ที่เกี่ยวข้อง ดังนั้น การแก้ไขจะทำได้ยาก**

Coupling เป็นการบอกถึงระดับความเกี่ยวข้องกันของต่าง ๆ ในระบบ หากระบบใดมีความเกี่ยวข้องกันของ Object ที่สูง ก็จะทำให้ระบบนั้นมีความอิสระน้อยลง

เช่น ใน Class ที่เราสร้างขึ้นมา มีการอ้างไปถึง Class อื่น ๆ อีก ถ้าเกิด Class นั้นมีการเปลี่ยนแปลงขึ้นมาก ก็จะส่งผลกระทบถึง Class ดังนั้น **ควรให้องค์ประกอบในระบบมีความเกี่ยวข้องกันน้อยที่สุด**

Coupling is a **measurement of the level of dependency between modules**. Low coupling is desired because it **increases the maintainability and readability of the module**. There are many types of coupling (shown lowest to highest)

Message coupling is when modules **send “message” back and forth to each other via a public exchange like web services** using SOAP (Simple Object Access Protocol) or client-server SQL interfaces. **This is the lowest form of coupling.**

Data Coupling is when modules **share data via parameters**.

Stamp Coupling is similar to data coupling but **passes a complete data structure**.

Control Coupling is when **one module controls the functionality of another**.

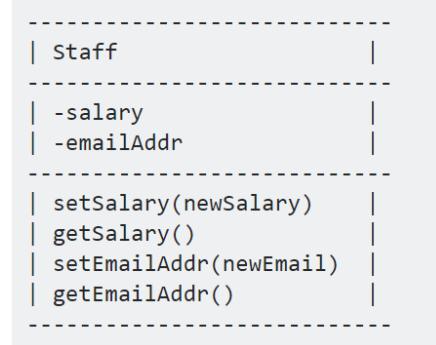
External Coupling is when modules **share a common interface determined by external criteria**.

Common Coupling is when **two modules share common global data variables**.

Content Coupling is when one module relies on the internal workings of another module. [This is extremely high coupling.]

Example of High Cohesion:

Example of Low Cohesion:



Case Study of Coupling Impact :

ตัวอย่าง

Class A มีการดึงข้อมูลราคาหุ้นจาก Web Service และ

Class B มีหน้าที่ในการหา Market Value ของ Portfolio

ผลกระทบ

เมื่อ Class A มีการเปลี่ยนแปลง โดยเพิ่ม Parameter เข้าไป
จะส่งผลให้ Class B ต้องเพิ่ม Parameter เข้าไปด้วย

ลักษณะของ Coupling มีอยู่ 2 ประเภทคือ

1. Procedural programming

2. Object-oriented programming (OOP)

1. Procedural programming

Pathological coupling (Content coupling)

คือ Module หนึ่ง เข้าไปแก้ไข Module อีก Module ได้ เช่น การเข้าไปแก้ไขตัวแปร

Global coupling (Common coupling)

คือ การแชร์ข้อมูล Global ร่วมกัน เช่น UserID ที่ใช้งานอยู่ปัจจุบัน

External coupling คือ Module แต่ละขึ้นอยู่กับข้อมูลที่มาจากภายนอก เช่น จาก WebService หรือ Text file ที่มี format เฉพาะตัว

Control coupling

คือ การส่งข้อมูลที่เป็นโครงสร้าง เช่น Object Student

1. Procedural programming (ต่อ)

Stamp coupling (Data-structured coupling)

คือ การส่งข้อมูลที่เป็นโครงสร้าง เช่น Structure/ Class Student

Data coupling คือ Module มีการแชร์ Data ร่วมกันผ่านทางพารามิเตอร์ (pass by value & pass by reference Data transfer)

Message coupling (low) คือ ความสัมพันธ์ของสอง Module นั้นเกิดจากการเรียกใช้งาน ส่งผ่าน parameter (pass by reference) และให้ผลลัพธ์เป็น ข้อความแสดงสถานะการทำงานต่าง ๆ

No coupling คือ ไม่มีความสัมพันธ์ใดๆเลย

2. Object-oriented programming (OOP)

Subclass Coupling

ความสัมพันธ์ระหว่าง SubClass กับ Super Class

ตัวอย่าง : Interface A มีความสัมพันธ์กับ Class B, C และ D

Temporal coupling

ใน Object มีกิจกรรม หรือหน้าที่ในการทำงานตั้งแต่ 2 อย่างเกิดขึ้นพร้อมกัน

ตัวอย่าง : Object E มี Method GetActiveUser กับ GetLastLogin ที่ถูกเรียกใช้งานพร้อมกันทุกๆ 1 วินาที

Class: Vehicle



คลาสรถยนต์ประกอบด้วย object หลาย object ด้วยกัน เมื่อนำ object ทุก object มาประกอบเพื่อให้เป็นรถยนต์ 1 คัน มีขั้นตอนในการประกอบที่ยุ่งยากมาก และเมื่อ มี object ที่เปลี่ยนมาก

เวลาแก้ไขเมื่อเกิดปัญหา ก็จะทำให้เกิดความยุ่งยาก เช่นกัน หรือที่เรียกว่าการเขียนโปรแกรม คือ “Coupling” สูง

<http://www.gurgeek.com/education/coupling-คืออะไร-และ-สำคัญอย่างไร/>
<http://naiwaen.debuggingsoft.com/2014/07/cohesion-vs-coupling/>

High cohesion

is desirable because modules written with high cohesion tend to be **more understandable, more reliable, and reusable.**

Typically, a highly cohesive module is performing a small part of a highly complex program.

Each module has a job, which, **if programmed correctly, will perform its function very well.**

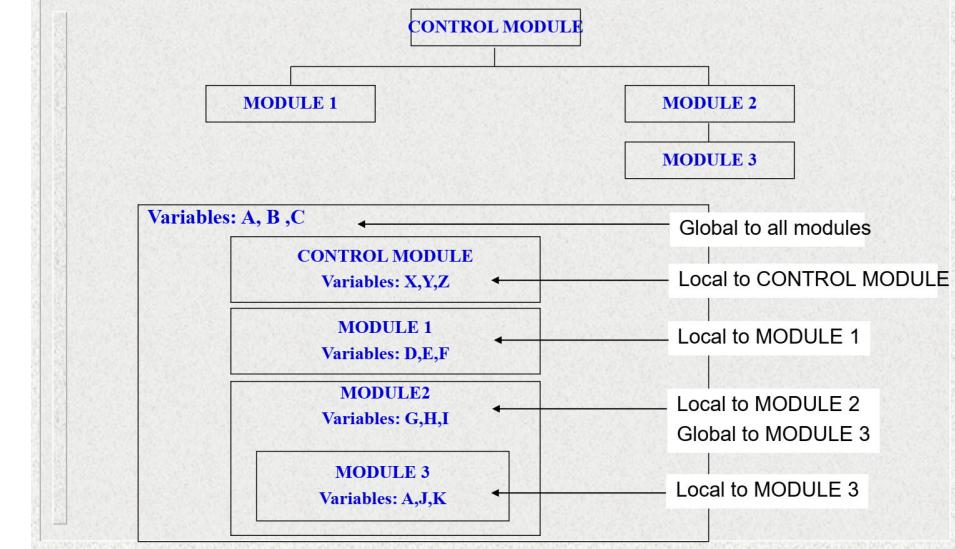
Local and Global Variables

แนวคิดตัวแปรที่ใช้งานแบบได้ดังนี้

- ❑ Local Variables
- ❑ Global Variables

ความแตกต่างระหว่างทั้ง 2 แบบคือ ขอบเขต (scope) ของการใช้งานตัวแปร

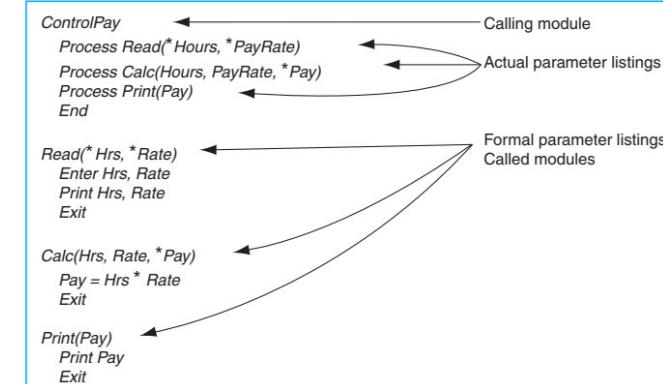
Local and Global Variables(con)



Parameters

- **Parameters** คือ ค่าข้อมูลที่ถูกส่งผ่านจากโมดูลหนึ่งไปยังโมดูลอื่นๆ สำหรับพารามิเตอร์เป็นรูปแบบหนึ่งของ coupling ที่ใช้สื่อสารระหว่างโมดูล
- **Formal parameter** เป็นรูปแบบการแสดงรายละเอียดเพื่อการประกาศ(declaration)โมดูล
- **Actual parameter** เป็นรูปแบบการแสดงในการเรียกใช้(calling)โมดูล

ชื่อของพารามิเตอร์ที่ใช้หั้งสองแบบไม่จำเป็นต้องเป็นชื่อเดียวกัน โดยค่าที่ส่งผ่านมาจะเรียงตามตำแหน่งในโมดูล



*Indicates call-by-reference parameter, those parameters without an asterisk are call-by-value parameters.

Parameters (con)

Parameters สามารถแบ่งออกได้เป็น 2 ประเภทคือ

- **Call-by-Value Parameter**
ในภาษาคอมพิวเตอร์เรียก pass by value
- **Call-by-Reference Parameter**
ในภาษาคอมพิวเตอร์เรียก pass by reference

Parameters(con)

Module 1
A = 3
B = 4
C = 5

① Process Module2 (A, B, *C)

EXIT

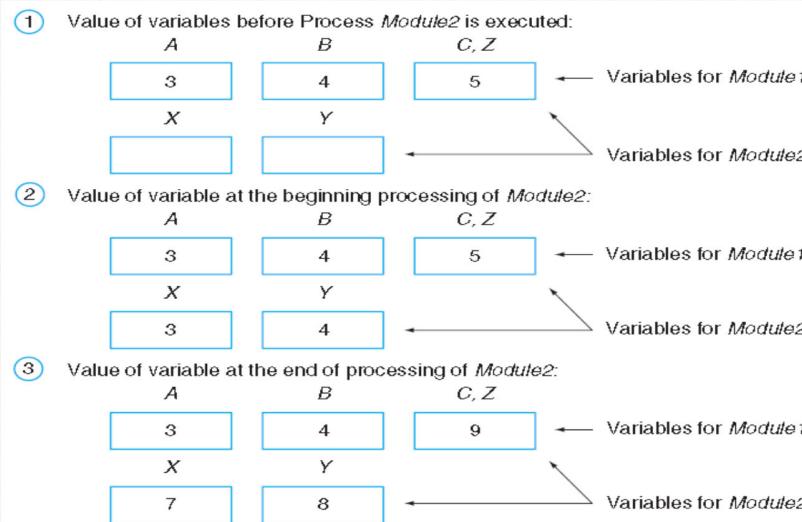
② Module2 (X, Y, *Z)

X and Y are call-by-value parameters
Z is a call-by-reference parameter

X = 7
Y = 8
Z = 9

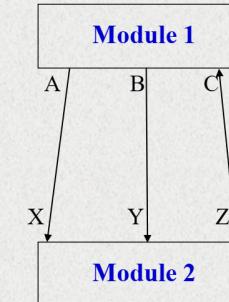
③ EXIT

Parameters(con)



Parameters(con)

พารามิเตอร์เป็นวิธีการหนึ่งที่ดีของ coupling module และไม่ต้องกังวลถึงการประกาศชื่อตัวแปรข้างนอกแต่ละโมดูล



Coupling Diagram

Return Values

- เป็นวิธีการหนึ่งของ coupling module
- โดยการคืนค่ากลับที่ชื่อฟังก์ชันได้เพียงค่าเดียว
- โดยต้องใช้คำสั่ง return ทำงานในโมดูลด้วย เพื่อทำการส่งค่ากลับ
- ตัวอย่าง

$$\text{SquareRoot} = \text{SQRT}(16) + 5$$

Variable Names and Data Dictionary

- Variable Name เป็นการกำหนดชื่อที่สัมพันธ์กับข้อมูลที่ไปแทน
 - ❖ ข้อมูลเกี่ยวกับชั่วโมง ก็จะใช้ชื่อตัวแปรเป็น HOURS
 - ❖ โดยในโปรแกรมหนึ่งความมีเพียงชื่อเดียว(unique)
- Data dictionary เป็นเครื่องมือที่ช่วยให้การเก็บรายละเอียดของตัวแปรข้อมูลที่ใช้งานในโปรแกรมทั้งหมด

Variable Names and Data Dictionary(con)

Data Dictionary

Item	Variable Name	Data Type	Module	Scope	Pseudonym/Module	Error Check
Hours worked	Hours	Numeric—real	ControlPay	Local	Hrs	None
Hours worked	Hrs	Numeric—real	Read/ Calc	Parameter	Hours	Hours < 0
Pay rate	PayRate	Numeric—real	ControlPay	Local	Rate	None
Pay rate	Rate	Numeric—real	Read/ Calc	Parameter	PayRate	Pay rate < 4.00
Net pay	Pay	Numeric—real	ControlPay	Local	None	None
Net pay	Pay	Numeric—real	Calc/ Print	Parameter	None	None

1. เชื่อหรือไม่ว่าการ “ดูหนังส่ายองขวัญ” กลายเป็นเหตุการณ์ลดน้ำหนักรูปแบบใหม่ไปเสียแล้ว ผลการศึกษาของมหาวิทยาลัย Westminister กรุงลอนדון ประเทศอังกฤษ ระบุว่าการดูหนังส่ายองขวัญ เป็นเวลา 90 นาที ช่วยเผาผลาญแคลอรี่ของเราราวๆ ประมาณ 113 แคลอรี่ต่อหนัง 1 เรื่อง เทียบเท่ากับการเดิน เป็นเวลา 30 นาที การเผาผลาญแคลอรี่ 7,700 จะลดน้ำหนักได้ 1 กิโลกรัม

ดร.วิชาวดีเล่าไว้ "เมื่อหัวใจทำงานเร็วขึ้น ปั๊มเลือดไปทั่วร่างกายเร็วขึ้น ร่างกายก็จะหล่อละเวนรีลีน อาย่างรวดเร็ว ตอนนี้เองที่เราจะสึกดีนั่นเด่นและหาดกล้า ในขณะที่ดูหนังส่ายองขวัญ ร่างกายจะก่อเกิด ความเครียดรุนแรง ในระยะเวลาสั้น ๆ ซึ่งอาการทั้งหมดนี้ จะส่งผลให้ลดความอยากอาหารและกระตุ้นการเผาผลาญแคลอรี่ได้สูงขึ้น" หากต้องการลดน้ำหนัก 3 กิโลกรัมใน 1 เดือน จะต้องดูหนังเฉลี่ยวันละกี่เรื่อง

สิ่งที่นักศึกษาต้องทำส่ง คือ

- PAC
- Interactivity (Structure) Chart
- IPO Chart
- Coupling Diagram และ Data Dictionary
- Flow Chart
- Algorithm โดย Track การทำงานเป็นรายบรรทัด พร้อมสถานะของค่าที่เก็บในแต่ละตัวแปร

The Three Logic Structures

Logic structures หรือโครงสร้างความคุม เป็นการควบคุมการให้ผลข้อมูลผ่านโมดูล มี 3 แบบ คือ

- The sequential logic structure**
- The decision logic structure**
- The loop logic structure**

Example Code

```
#include <iostream >
using namespace std;
// function Prototype
void Read(int &Hours, int &PayRate);
void Calculate(int Hours, int PayRate, int &GrossPay);
void Print(int GrossPay);
int main()
{
    int Hours, PayRate, GrossPay;
    Hours = PayRate = GrossPay = 0;
    Read( Hours, PayRate);
    Calculate( Hours, PayRate, GrossPay);
    Print( GrossPay);
    return(0);
}
```

Example Code(con)

```
void Read(int &Hours, int &PayRate)
{
    cout << "Enter Hours and Payrate : ";
    cin >> Hours >> PayRate;
    cout << endl;
}

void Calculate(int Hours, int PayRate, int &GrossPay)
{
    GrossPay = Hours * PayRate;
}

void Print(int GrossPay)
{
    cout << "Employee have grosspay = " << GrossPay << endl;
}
```

Question ?

Auto Vending Machine

Develop six Step problem solving

Change Money with fewest return of Thai Banknote and coin after Purchase product that show label price with exclude vat. must include vat before change. (500, 100, 50, 20, 10, 5, 2, 1)

[1] PAC	[2] Interactivity Chart	[3] IPO	500
[4] Coupling Diagram	[5] Data Dictionary		100
[6] Flowchart	[7] Algorithm		50
[8] Step by Step Tracking Code			20

Example

Purchase: 123 10

Pay: 502 5

Output PriceVat: 132 2

Change: 370 1

500=0, 100=3, 50=2, 20=1, 10=0, 5=0, 2=0, 1=0 1