



www.siam2dev.com



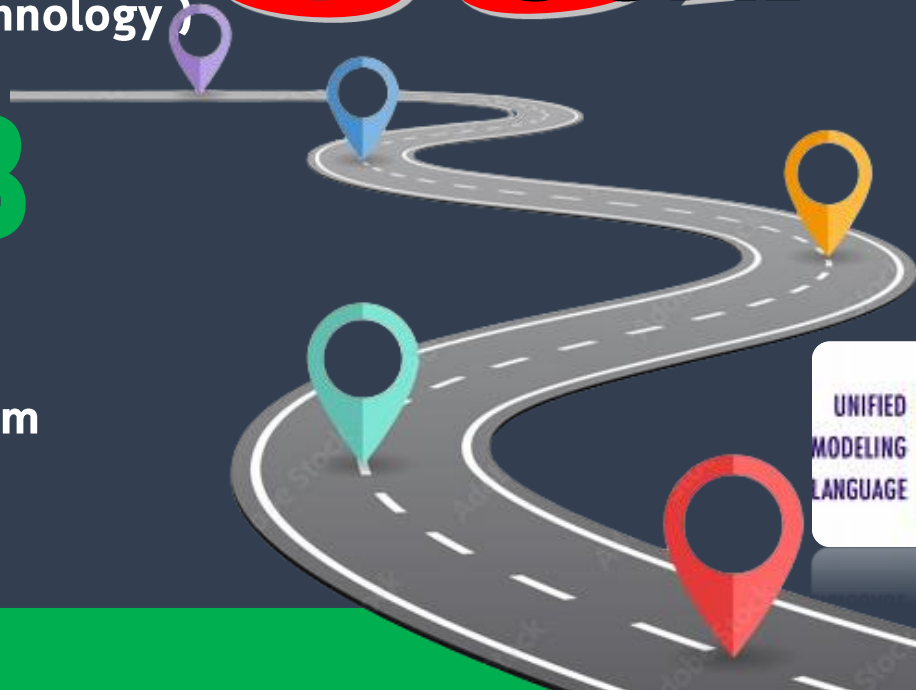
วิชา การวิเคราะห์และออกแบบระบบเชิงวัตถุ (4122506)

บทที่ 1 หลักการเชิงวัตถุเบื้องต้น
(Concept of Object-Oriented Technology)
By

1/2568

ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่องเนียม
Asst. Prof. Dr. Nattapong Songneam
เว็บไซต์ :: <http://www.siam2dev.com>
อีเมล :: xnattapong@gmail.com

OOAD



Last Update : 24/06/2568

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

คำอธิบายรายวิชา

วัฏจักรของการพัฒนาระบบงาน แนวคิดเกี่ยวกับเทคโนโลยีเชิงวัตถุ หลักการพัฒนาระบบงานแบบอินครีเมนตัน หลักการใช้ภาษาทางภาพเพื่อการออกแบบ UML (Unified Modeling Language) ขั้นตอนการวิเคราะห์ระบบ ประกอบด้วยการสร้างแบบจำลองเกี่ยวกับความต้องการการใช้ แผนภาพ Use Case การคิดและการวิเคราะห์เชิงนามธรรม การสร้างแบบจำลองเชิงวิเคราะห์ แนวคิดการค้นหาวัตถุ แนวทางการออกแบบเชิงวัตถุ การออกแบบคลาสการออกแบบความสัมพันธ์ระหว่างคลาสและวัตถุ การสร้างโปรแกรมด้วยวิธีใช้ CASE Tools แนวคิดเกี่ยวกับการทดสอบโปรแกรมตามแนวข้อกำหนด

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

ทบทวน (Reviews)

1. Algorithm

2. Flow Chart

3. Programming Language

4. ERD

5. Database

6. Information System Analysis and Design

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

ทบทวน (Reviews)

- *Algorithm*
- *SDLC: Software Development Life Cycle*
- *Overview of Object-Orientation*

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

Reviews

Algorithm : ความหมาย

ความหมายของขั้นตอนวิธี

อัลกอริทึม (อังกฤษ: algorithm) หมายถึง กระบวนการแก้ปัญหาที่สามารถเข้าใจได้ มีลำดับหรือวิธีการในการแก้ไขปัญหาคิดปัญหาหนึ่ง**อย่างเป็นขั้นเป็นตอนและชัดเจน** เมื่อนำเข้าอะไรแล้วจะต้องได้ผลลัพธ์เช่นไร ซึ่งแตกต่างจากการแก้ปัญหาแบบสามัญสำนึก หรือฮิวริสติก (heuristic)

- การเขียนโปรแกรมภาษาคอมพิวเตอร์ 1
- การเขียนโปรแกรมภาษาคอมพิวเตอร์ 2

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การนำเสนอขั้นตอนวิธี

อัลกอริทึม สามารถนำเสนอให้ผู้อื่นเข้าใจได้หลายวิธีดังนี้

PSEUDO-CODE (รหัสจำลอง/รหัสเทียม)

FLOW CHART (ผังงาน)

STRUCTURE CHART (ผังโครงสร้าง)

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การนำเสนอขั้นตอนวิธี

START

START

START

1. START
2. ...
3. INPUT
 1. READ /ENTER /INPUT
 2. Initialization Stage / Declaration / Definition Stage
4. PROCESS
 1. COMPUTE / ASSIGN VALUE / CALCULATION / INCREASE / DECREASE
5. OUTPUT
 1. PRINT / OUTPUT / DISPLAY / SHOW MESSAGE / RESULT
6. STOP

STOP

1

บทที่ 1

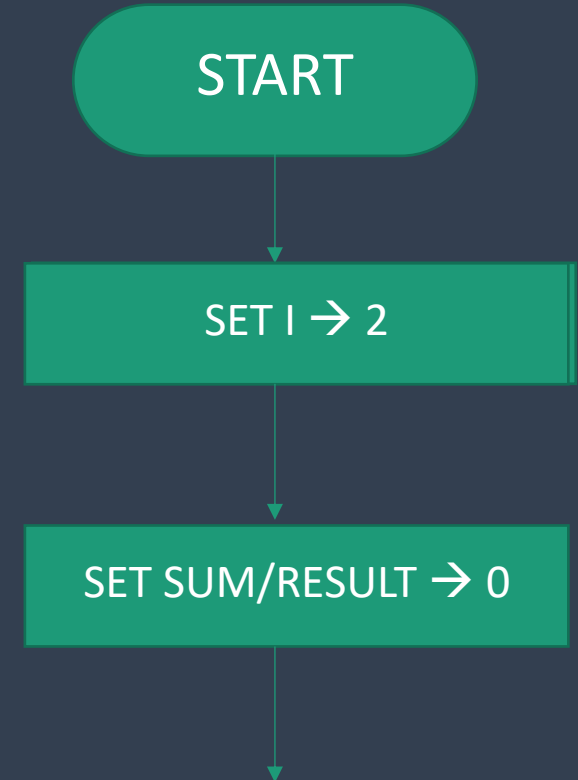
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การนำเสนอขั้นตอนวิธี

$2 + 4 + 6 + ..1000000$

- HAVE NOT INPUT
- LOOP ,while , do , for
- FOR initial TO Finalize Clear Memory
- FOR SET I $\rightarrow$ 2 TO 1000000 ~~1,000,000~~
-
- END FOR



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การนำเสนอขั้นตอนวิธี

Main Program

X =Pow(2,3)

Sub Routine

Math.Pow(2,3)

Function
Procedure

X = 8

- Standard Function ฟังก์ชันหลักที่ภาษาคอมพิวเตอร์สร้างไว้ให้เรียกใช้งานได้
- User Define Function โปรแกรมเมอร์ต้องเขียนขึ้นมาเอง

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น
(Concept of Object-Oriented Technology)

Reviews

Algorithm : การนำเสนอขั้นตอนวิธี

Main Program

- public double calTotalProduct(proPrice, proQTY) {
- x = 0;
- }

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การนำเสนอขั้นตอนวิธี

Sub routine

```
Public Function calTotalProduct1(ByVal proPrice, ByVal proQTY) As Double
```

```
    Return proPrice * proQTY
```

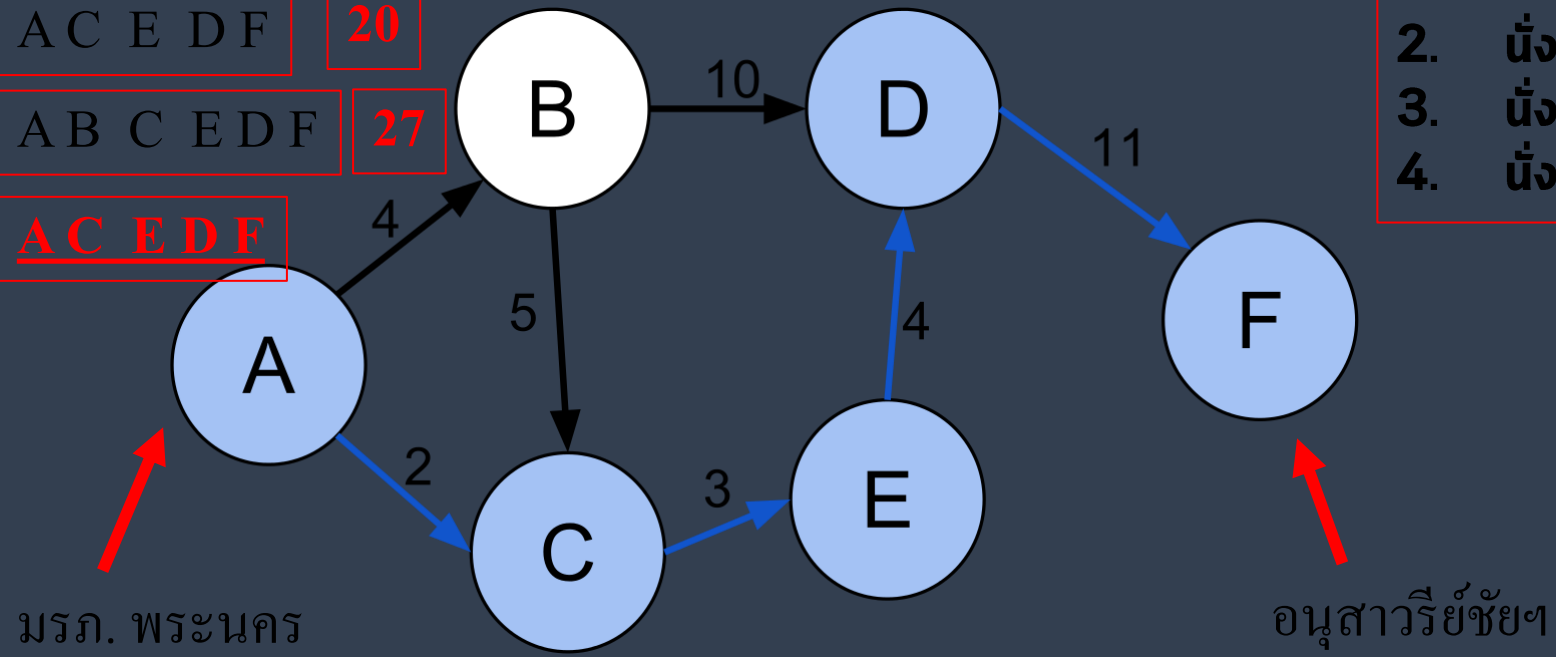
```
End Function
```

None case-sensitive

การหาเส้นทางที่สั้นที่สุด (shortest path)

- AB D F 25
- AC E D F 20
- AB C E D F 27
- AC E D F

- วิธีการ
1. นั่งแท็กซี่
 2. นั่งรถไฟฟ้า
 3. นั่งรถส่วนตัว
 4. นั่งรถประจำทาง



Graph Theory / Tree

Salesman Transversal
Google Map , Router

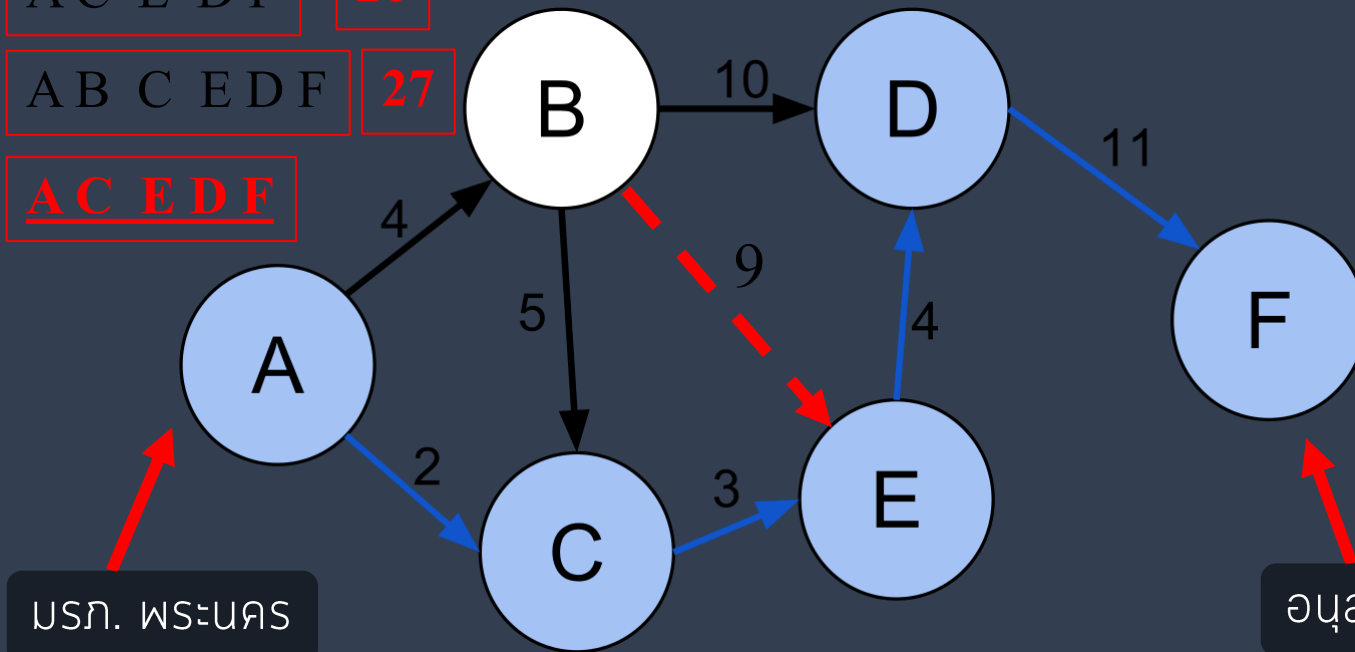
การหาเส้นทางที่สั้นที่สุด (shortest path)

AB D F 25

AC E DF 20

AB C EDF 27

AC EDF



วิธีการ

1. นั่งแท็กซี่
2. นั่งรถไฟฟ้า
3. นั่งรถส่วนตัว
4. นั่งรถประจำทาง

มรท. พระนคร

อนุสาวรีย์ชัยฯ

Graph Theory / Tree

Salesman Transversal

แนวทางของการพัฒนาซอฟต์แวร์ (paradigm)

- การเขียนโปรแกรมแบบดั้งเดิม / แบบโครงสร้าง
 - *Traditional Software Development*
 - *Structural Software Development*
- การเขียนโปรแกรมแบบเชิงวัตถุ / การพัฒนาระบบเชิงวัตถุ
 - *UP, OOAD, OOSE, OOA, OOD, OOP, OO Testing*

Traditional

OO



Structural Programming

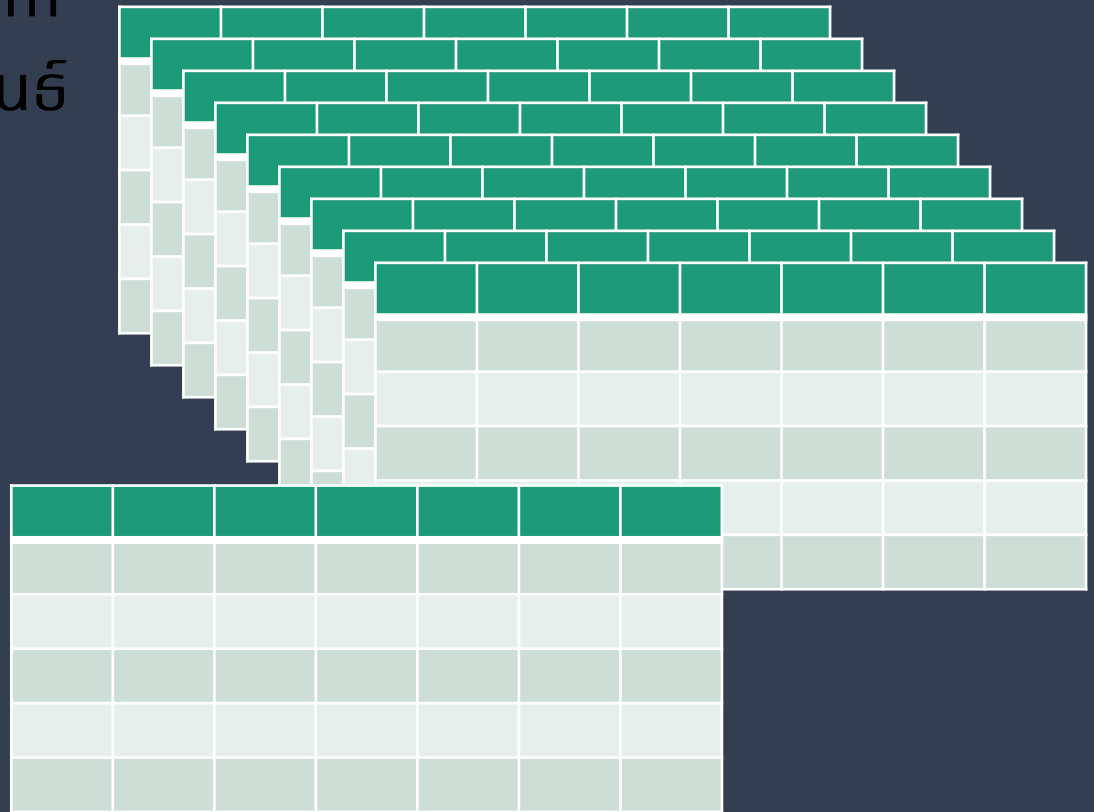
Basic, c, pascal,

PSEUDOCODE, FLOWCHART

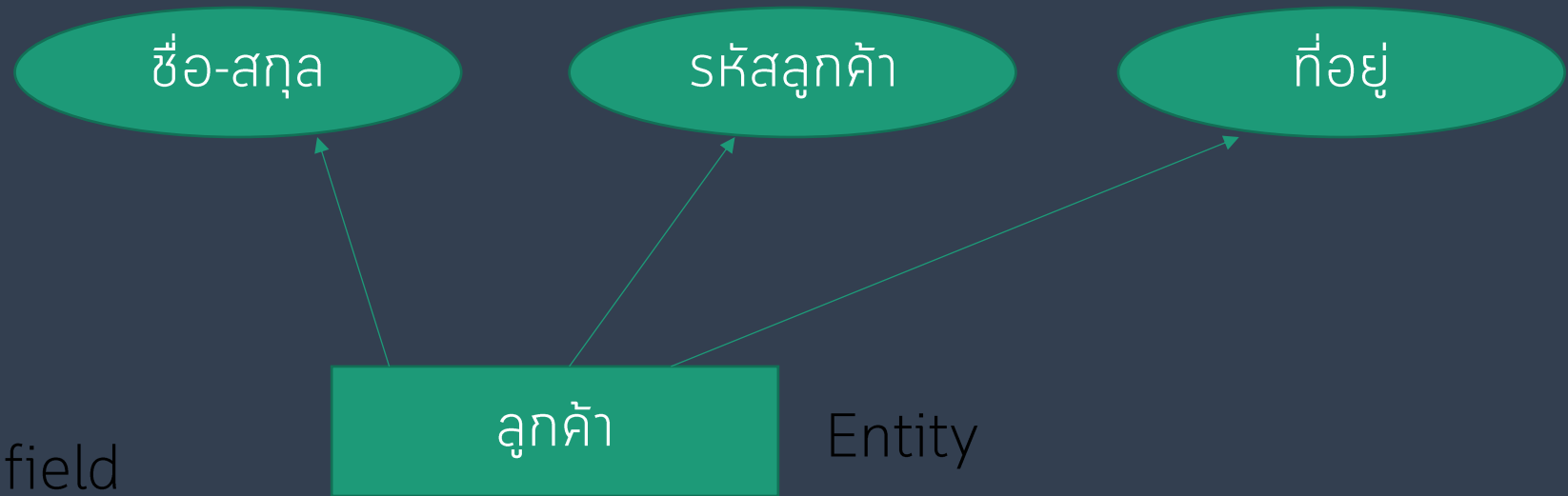
DED, EDD, NE

การออกแบบและจัดการฐานข้อมูล

- Relational database management system
- ระบบฐานข้อมูลเชิงสัมพันธ์
 - MS-SQL server
 - MySQL
 - Access ไม่สมบูรณ์
 - Oracle
 - DB2
 - Informix
 - MangoDB
 - SQLite
 - ฯลฯ



attribute



field

Entity

Table

CustID	CustName	CustAddress						
p0000001	Computer	เครื่องคอมพิวเตอร์ HP สีแดง						
	Printer							
	Speaker							
	Mouse							

record

Relational Database

- จงเขียนคำสั่ง SQL เพื่อแสดงรายการสินค้าที่มีราคามากกว่า 20000 บาท
- จงเขียนคำสั่ง SQL เพื่อแสดงรายชื่อสินค้าที่ขึ้นต้นด้วย C

TbProducts

ProID	ProName	ProDetail	ProFactory	ProQTY	ProType	ProIMG	ProPrice	ProUnit
p0000001	Computer	เครื่องคอมพิวเตอร์ HP สีแดง	HP	100	t01	Hp01.jpg	35000.75	u1
	Printer							
	Speaker							
	Mouse							

TbUnit

unitID	Detail		

TbProductType

TypeID	Detail	
t01	เครื่องคอมพิวเตอร์ ห้างเดอะ	

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

SQL : Structure Query Language

- จงเขียนคำสั่ง SQL เพื่อแสดงรายการสินค้าที่มีราคามากกว่า 20,000 บาท
- จงเขียนคำสั่ง SQL เพื่อแสดงรายชื่อสินค้าที่ขึ้นต้นด้วย C

```
SELECT * FROM TbProducts WHERE proPrice > 20000
```

```
SELECT ProName FROM TbProducts WHERE proName Like 'C%'
```

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

- SQL : Structure Query Language
 - DDL : Data Definition Language
CREATE TABLE , CREATE DATABASE
 - DML : Data Manipulation Language
INSERT , UPDATE , DELETE , SELECT
 - DCL : Data Control Language
GAINT , PERMISSION

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

ตารางที่ 1 ตารางข้อมูลลูกค้า Customer table

No.	Attribute name	Description	Data type	Key type	Reference
1	custID	รหัสลูกค้า	char(10)	PK	
2	Name	ชื่อลูกค้า	varchar(50)		
3	Address	ที่อยู่	varchar(50)		
4	Tel	เบอร์โทรศัพท์	varchar(50)		
5	cardID	หมายเลขประจำตัวประชาชน	char(13)		
6	taxID	หมายเลขประจำตัวผู้เสียภาษี	char(13)		
7	birthdate	วันเกิด	date		

พจนานุกรมข้อมูล Data Dictionary

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

จงเขียน Data Dictionary ของตารางข้างล่างนี้

ProID	ProName	ProDetail	ProFactory	ProQTY	ProType	ProIMG	ProPrice	ProUnit
p0000001	Computer	เครื่องคอมพิวเตอร์ HP สีแดง	HP	100	t01	Hp01.jpg	35000.75	u1
	Printer							
	Speaker							
	Mouse							

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

พจนานุกรมข้อมูล(Data Dictionary)

ชื่อตาราง	ลำดับ	ชื่อฟิลด์	ชนิด	ขนาด	KEY	หมายเหตุ
TbProducts	1	ProID	Text / Varchar	7	PK	รหัสสินค้า
	2	ProName	Text	50	-	ชื่อสินค้า

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

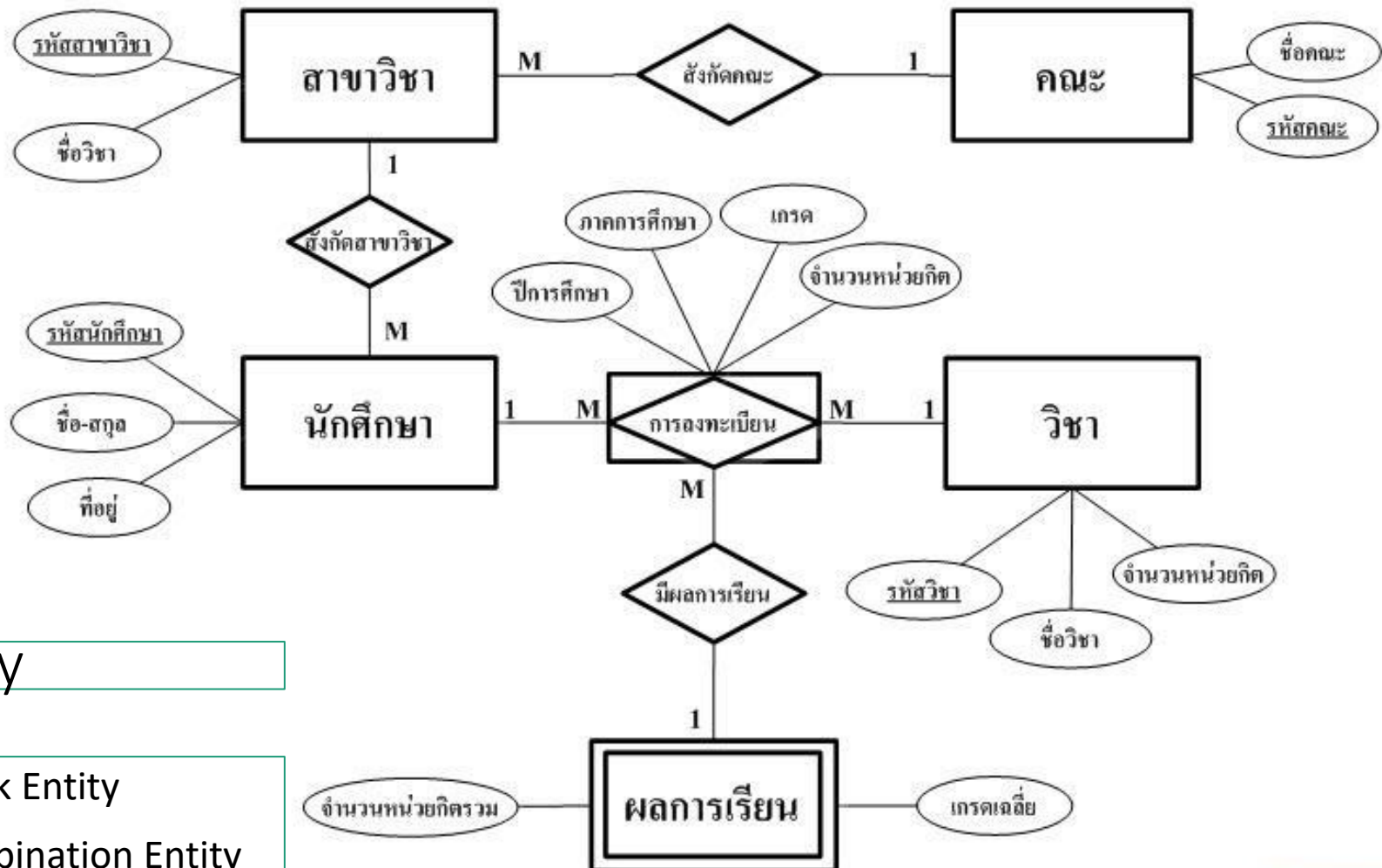
- จงค้นหาลูกค้าที่อยู่ กรุงเทพฯ
- จงค้นหาลูกค้าที่ เกิด ก่อนวันที่ 20/7/2562

Database1 - ฐานข้อมูล- C:\Users\COMIT\Documents\Database1.accdb (รูปแบบไฟล์ Access 2007 - 2013) - Access (การเปิดใช้งานผลิตภัณฑ์)

custID	name	address	tel	cardID	birtDate	คลิกเพื่อเพิ่ม
c001	นัฐพงศ์ ส่งเนียม	กรุงเทพ	09908744	11111111	27/7/2562	
c002	ณัฐพล ทุนมาดา	เชียงใหม่	07888888	22222222	1/8/2562	
c003	นที ขวัญดี	สงขลา	0444555	3333333	19/7/2562	
c004	นภัส มากมี	กรุงเทพ	087777	4444444	10/7/2562	

SELECT * FROMWHERE

ERD: Entity Relationship Diagram



Entity

- Week Entity
- Combination Entity



Entity

- Week Entity
- Combination Entity

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

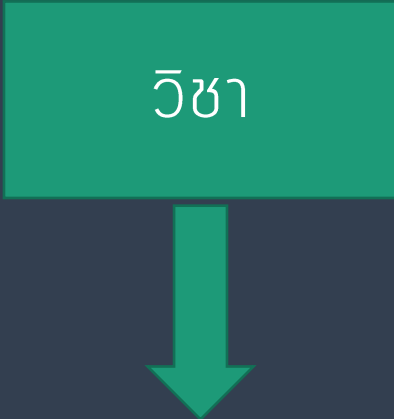
Reviews

Algorithm : การออกแบบและจัดการฐานข้อมูล

ER-Mapping

Relations หมายถึง ตาราง

วิชา



วิชา (รหัสวิชา , ชื่อวิชา , หน่วยกิต , คำอธิบายรายวิชา)

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : Programming Language + Database + SQL

รับชำระเงิน ค่าสมาชิก : ปรับปรุงข้อมูล

รับชำระเงิน ค่าสมาชิก
บันทึกการชำระเงิน ค่าสมาชิก

เลขที่เอกสาร	RP5610170000001	หมายเหตุ	
วันที่เอกสาร	17 / 10 / 2556 15 10 : 44		
รหัสสมาชิก	19992008590101751214 ...		
ชื่อสมาชิก	นางสาวนภาพรรณ ใจดี ...		
ประเภทสมาชิก	EXMT12 ...	ผู้รับเงิน	DEMO ...
	ค่าสมัครสมาชิก 1 ปี (12 เดือน)		คุณผู้ทดลองใช้งาน ระบบ
ราคา	1,000	พนักงาน	DEMO ...
ยอดชำระ	1,000		คุณผู้ทดลองใช้งาน ระบบ
		เข้าบัญชี	400-1 ...
			รายได้ขายสินค้า/บริการ

[สถานะภาพสมาชิก]

ตั้งแต่วันที่ 1 ตุลาคม 2556 15

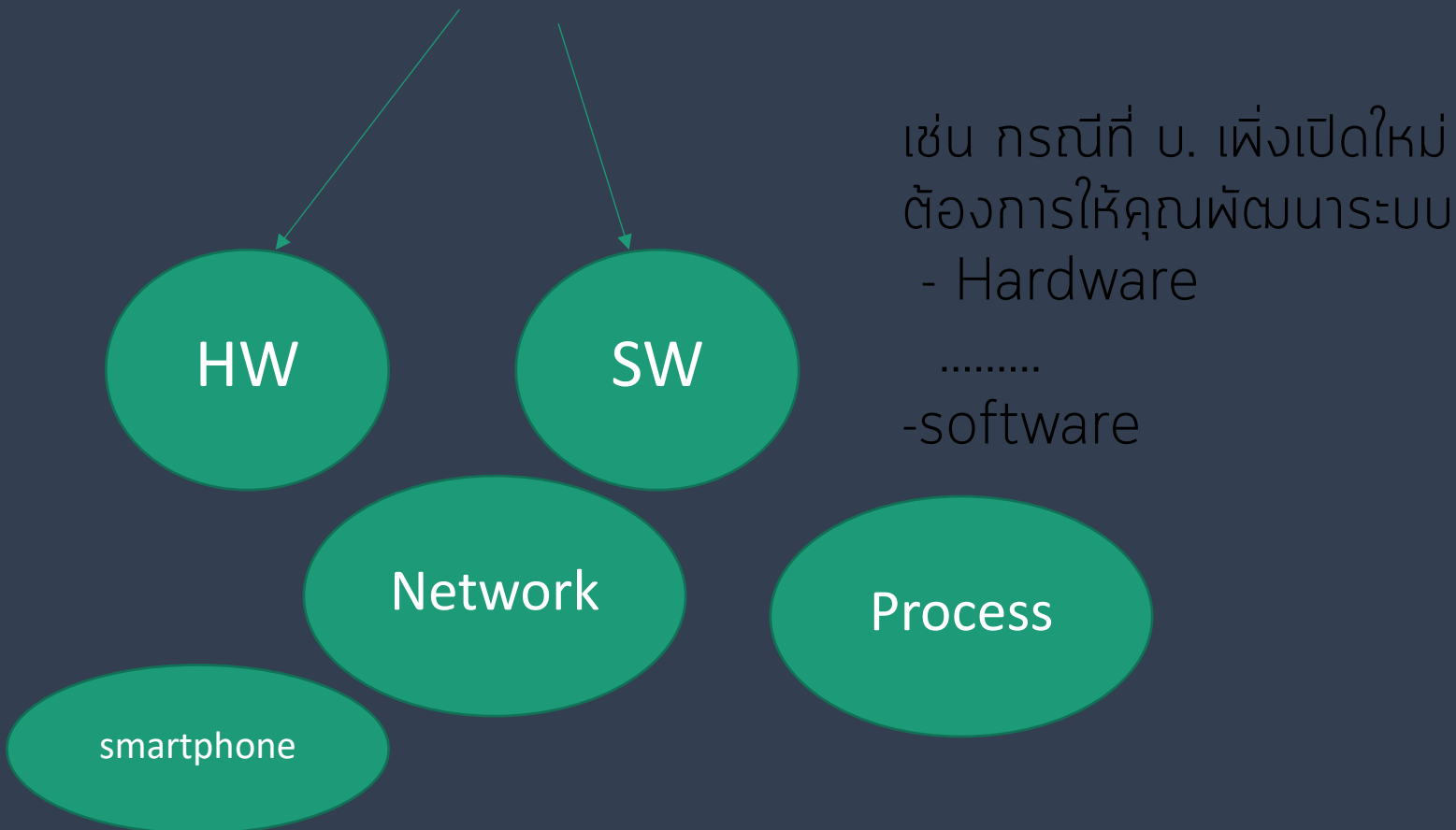
วันหมดอายุ 31 ตุลาคม 2557 15

 บันทึก  ยกเลิก

ลิขสิทธิ์ © 2013-09-26

ข้อแตกต่าง ระหว่างการพัฒนาโปรแกรม กับ การพัฒนาระบบ

- พัฒนาโปรแกรมก็มีขั้นเดียว ..เขียนโปรแกรม ...
- การพัฒนาระบบ ...ครบวงจร มากกว่า มีขอบเขตกว้างกว่า



ขั้นตอนในการพัฒนาโปรแกรมหรือการเขียนโปรแกรม

1. กำหนดปัญหา/วิเคราะห์ปัญหา
 2. การวิเคราะห์และออกแบบอัลกอริทึม
 3. พัฒนาโปรแกรม ลงรหัส
 4. ทดสอบโปรแกรม
 5. ติดตั้งใช้งาน
 6. ทำเอกสารประกอบโปรแกรม
 1. เอกสารสำหรับผู้ใช้โปรแกรม user manual
 2. เอกสารทางเทคนิค หรือเอกสารสำหรับผู้พัฒนา Technical document/manual
 7. การบำรุงรักษา ปรับปรุง
1. การเขียนโปรแกรมและขั้นตอนวิธี
 2. การเขียนโปรแกรมเชิงภาพ
 3. การเขียนโปรแกรมเชิงวัตถุ (OOP)

SDLC: Software Development Life Cycle

วิชา การเขียนโปรแกรมและขั้นตอนวิธี

Algorithm + Syntax

เรียน algorithm

เรียน การวิเคราะห์ปัญหา

การเขียนรหัสเทียม/รหัสจำลอง PSEUDO-CODE

เรียนการเขียนผังงาน flowchart , structure chart

เรียน คำสั่งเบื้องต้น

การทำเอกสารประกอบโปรแกรม

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : 2 วิชาที่เกี่ยวกับการวิเคราะห์และออกแบบระบบ

2 วิชาที่เกี่ยวกับการวิเคราะห์และออกแบบระบบ

- SAD : System Analysis and Design
- OOAD : Object - Oriented Analysis and Design

การรวบรวมข้อมูล

DFD : Data Flow Diagram

Process Description

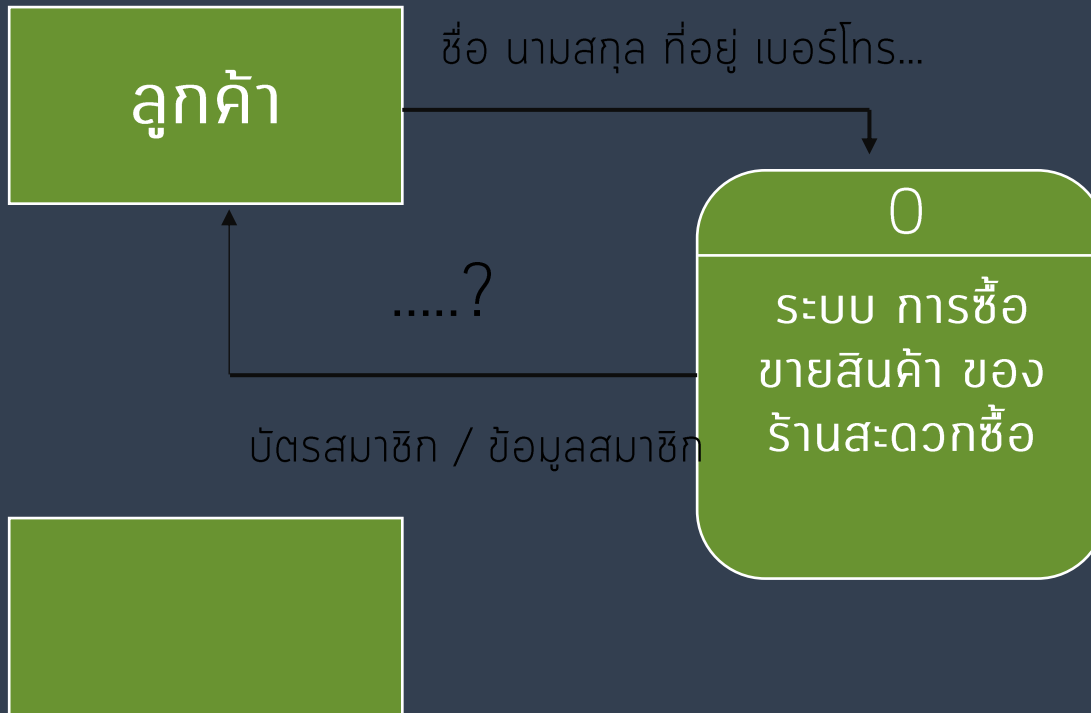
1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : DFD : Data Flow Diagram



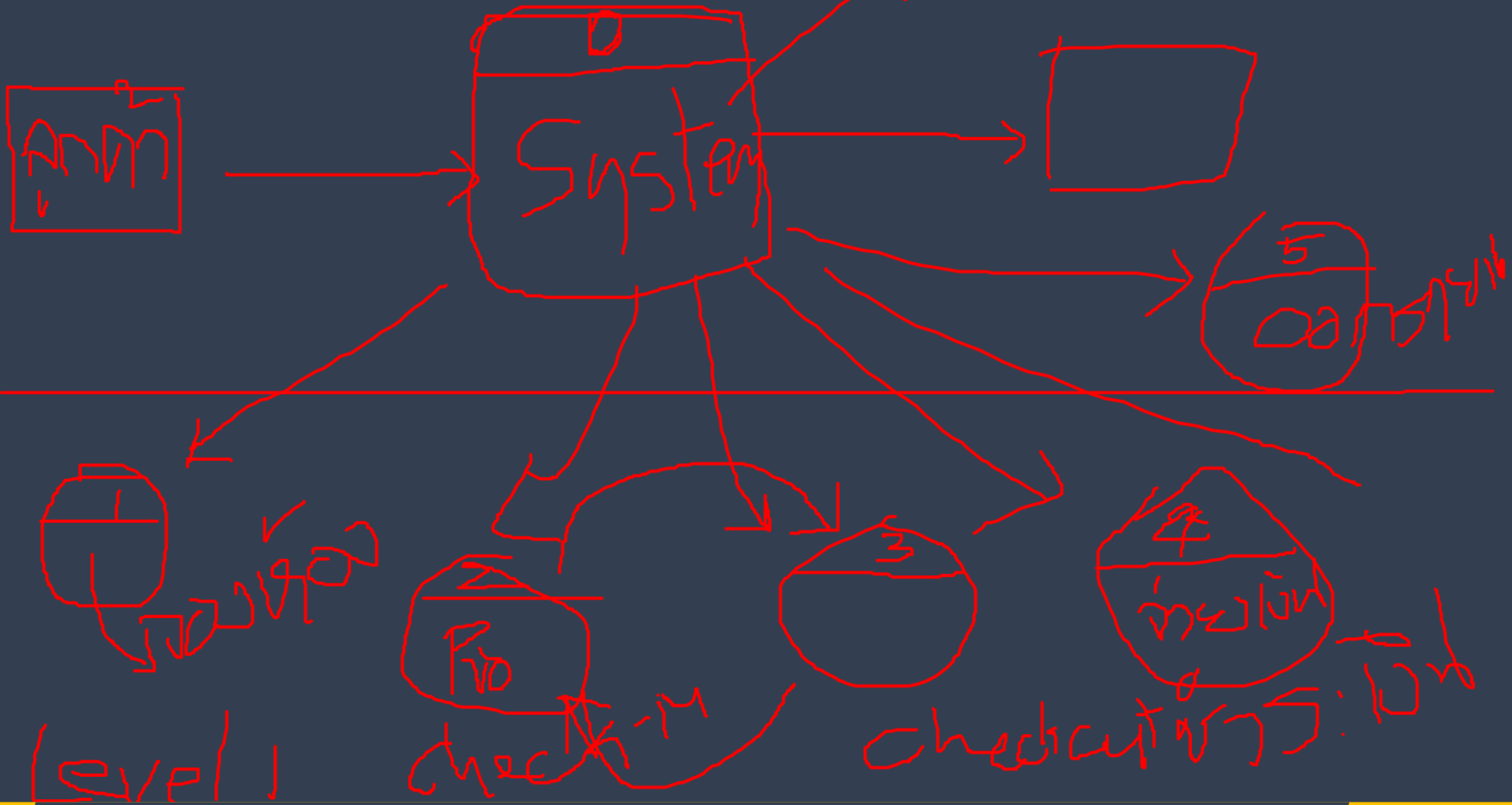
- ก. Entity
- ข. Boundary
- ค. External Source

Email: siam2dev@Hotmail.com
ชื่องาน
ชื่อนักศึกษา สหส.....
สาขา
รุ่น / หมู่

Context diagram ของระบบ การซื้อขายสินค้า ของร้านสะดวกซื้อ ABC

Level 0

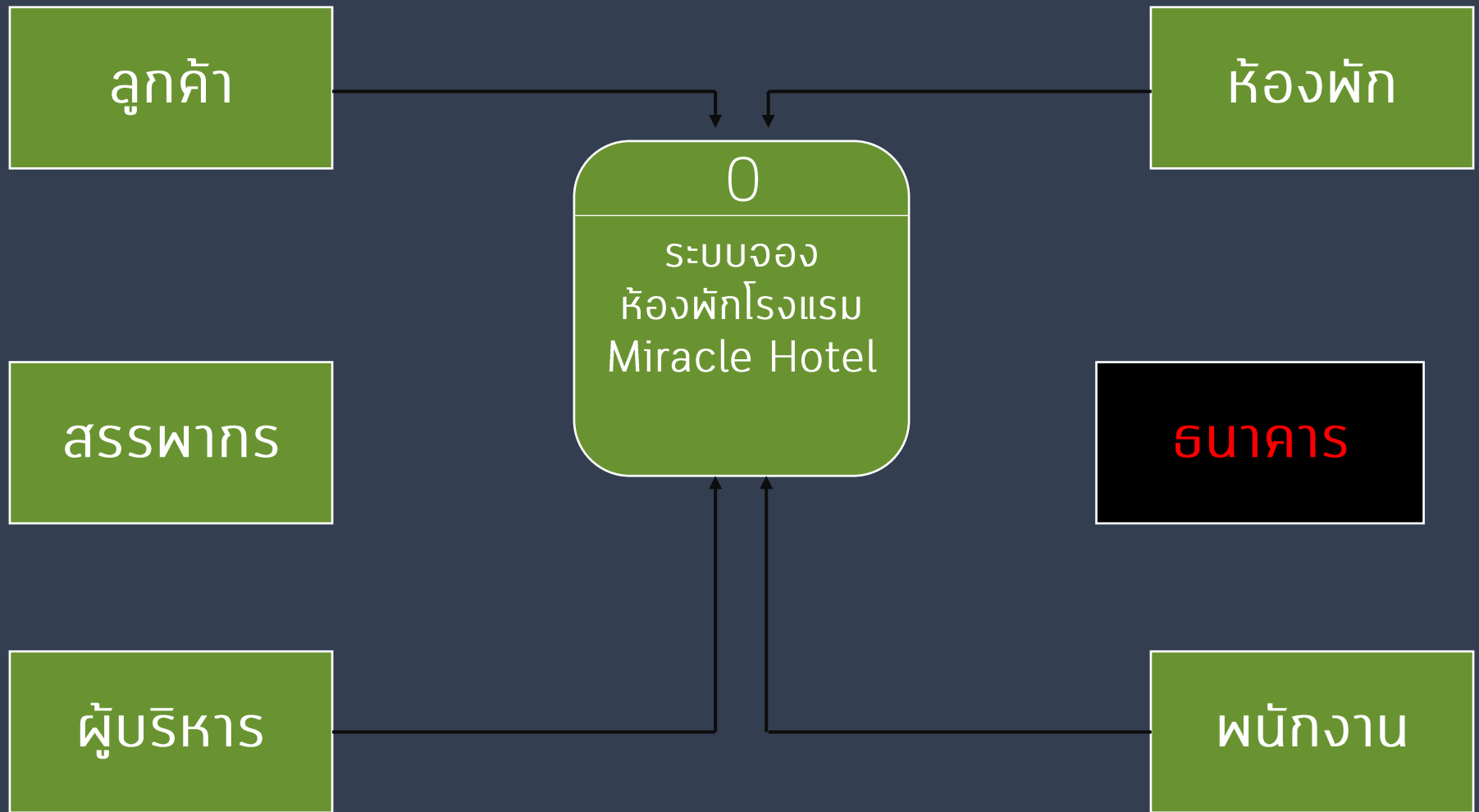
7-5
กำลังสินค้า (ใช้ 500 รส)



- Desktop-Application
- Web Application
- Mobile Application



- Desktop-Application
- Web Application
- Mobile Application

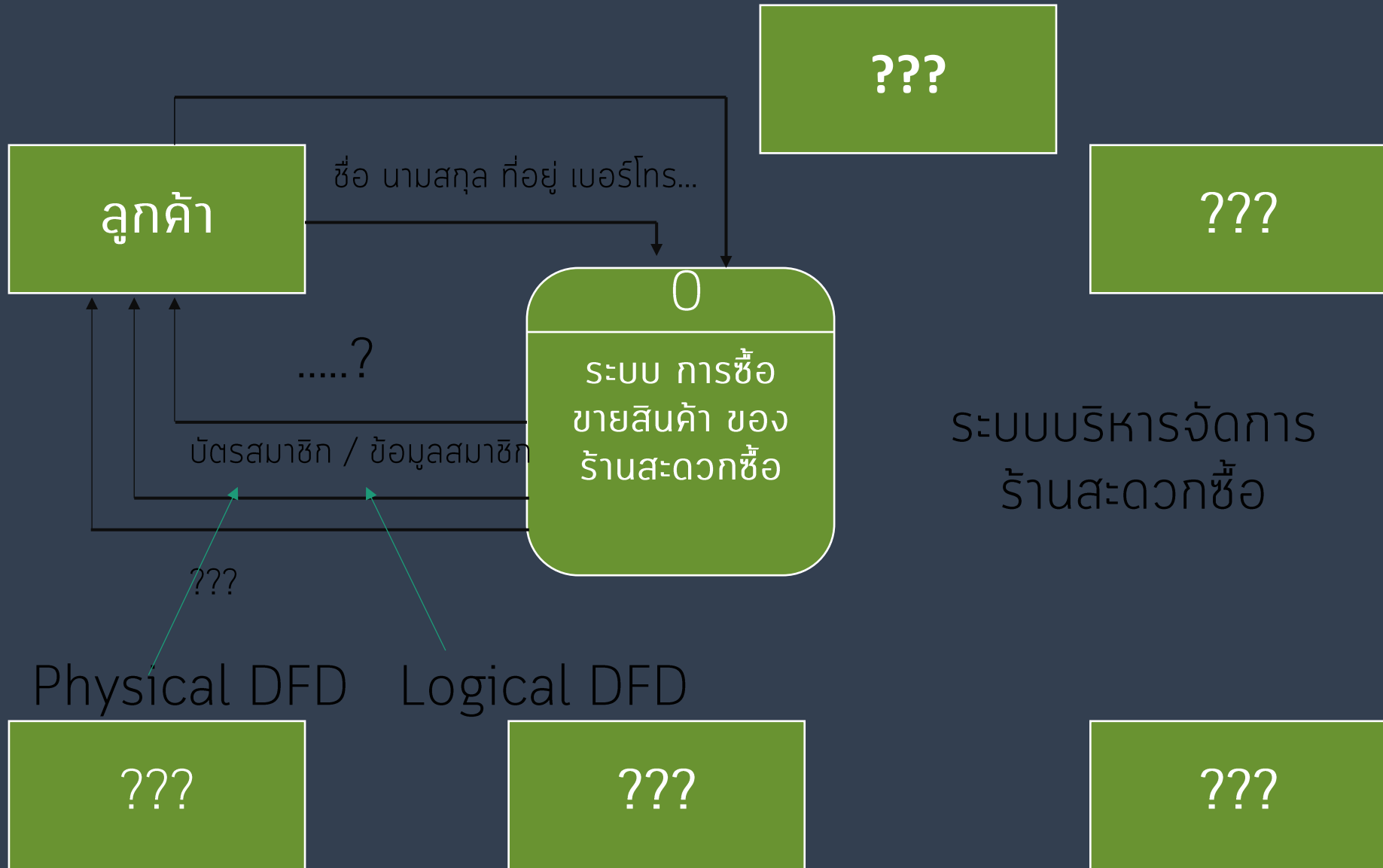




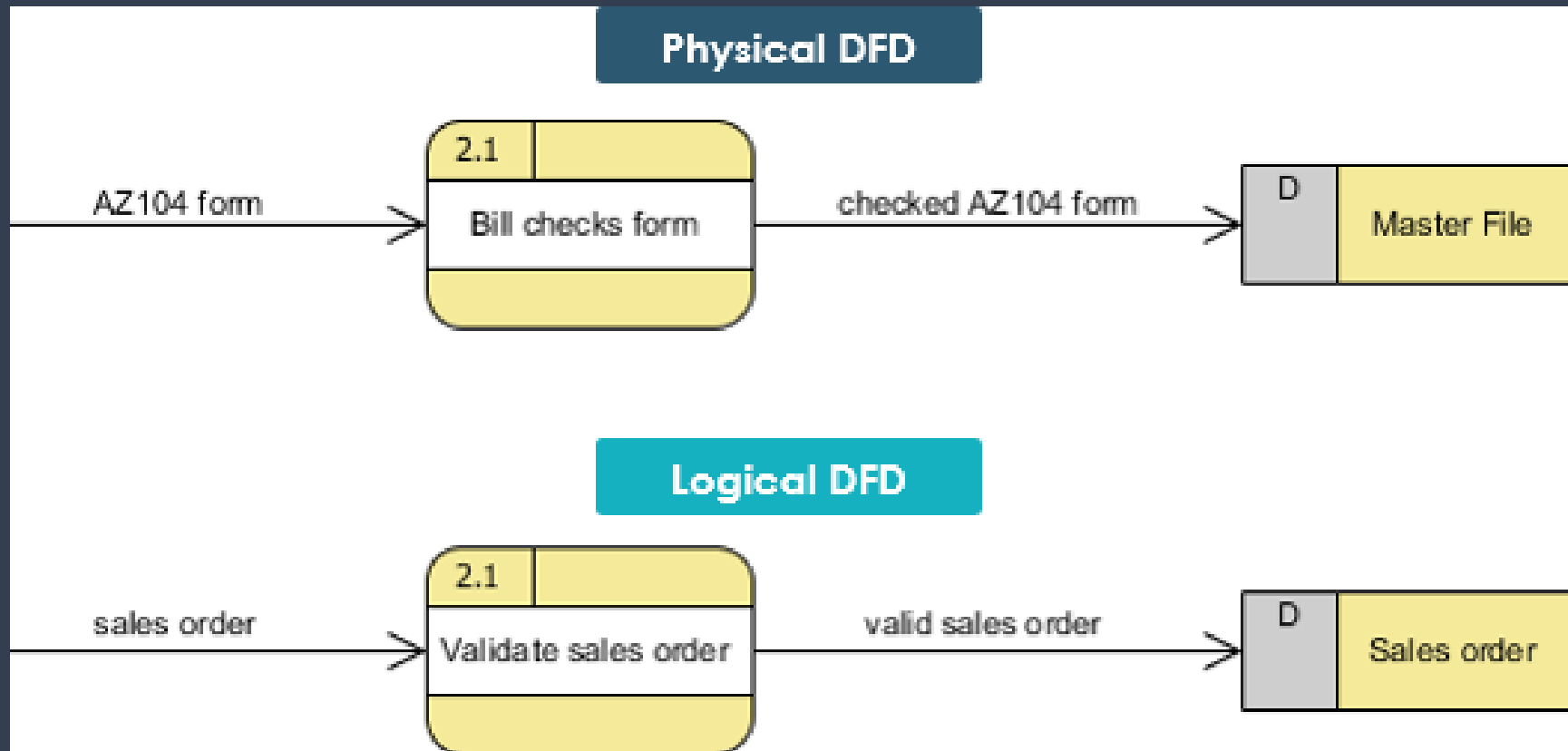
ข้อมูลการชำระเงิน

- Desktop-Application
- Web Application
- Mobile Application

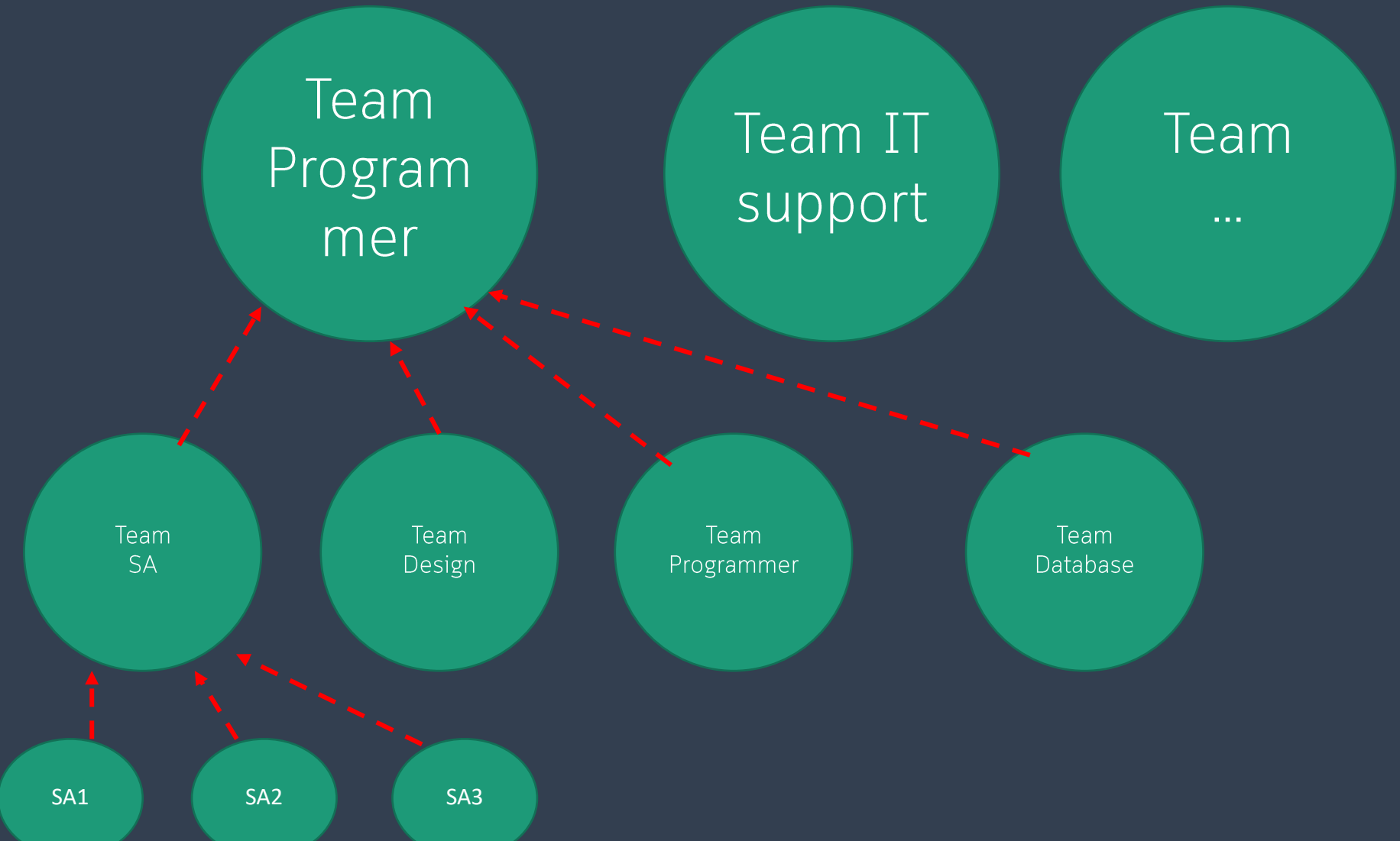




physical DFD vs logical DFD



โครงสร้างฝ่าย IT (ขึ้นอยู่กับแต่ละองค์กร)



Reviews

Algorithm : Traditional Software Development

- Structural Programming Language

- C,Pascal
- Basic

Text Mode/Dos >> Sequential >> Algorithms



SETUP

Algorithms คืออะไร ?

1. คือ : ขั้นตอนวิธี กระบวนการแก้ปัญหาที่อธิบายมาเป็นขั้นตอนที่ชัดเจน

3. สรุป ได้

2. ยก ต.ย. ได้



Reviews

Algorithm : วิธีการประมวลผล algorithm

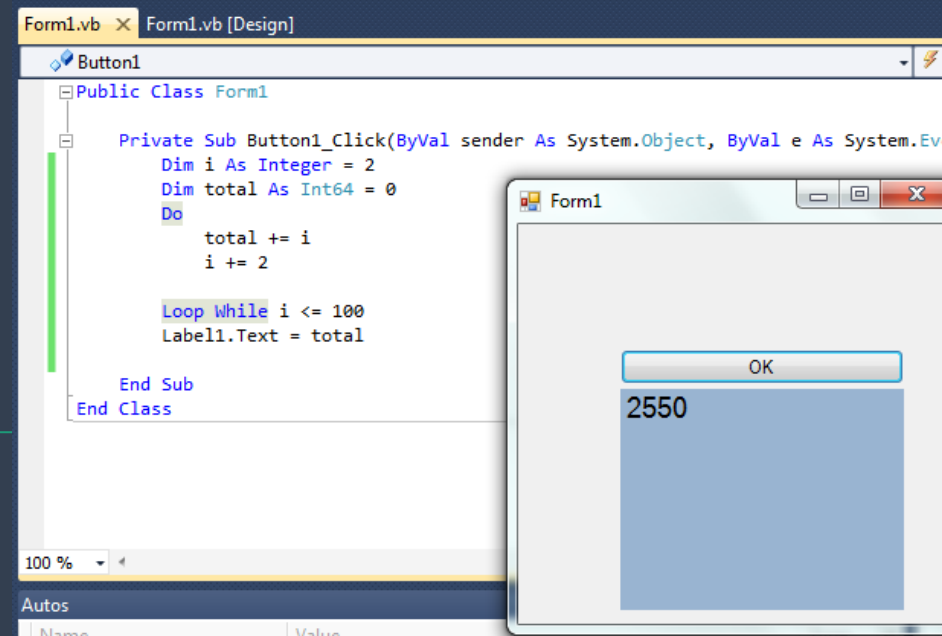
- เริ่มต้น
-การรับค่า x 2 4 6
- การประมวลผล
 - การทำลูป
 - กำหนดค่าเริ่มต้นให้ i เป็น 2
 - **SET I $\leftarrow$ 2**
 - **ทำ**
 - **Total $\leftarrow$ Total + I**
 - **เพิ่มค่า I $\leftarrow$ I + 2**
 - **ถ้า I $\geq$ 100 ให้ไปข้อต่อไป**
- การแสดงผล พิมพ์ Total ไปที่หน้าจอ
- สิ้นสุด

Reviews

Algorithm : วิธีการประมวลผล algorithm

1. START
2. SET Total $\leftarrow$ 0
3. SET I $\leftarrow$ 2
4. DO
 1. COMPUTE Total $\leftarrow$ Total + I
 2. Increase I $\leftarrow$ I + 2
 3. if (I $\geq$ 100) ให้ไปข้อต่อไป
5. DISPLAY "Total : " + Total
6. STOP

การปรับปรุงอัลกอริทึม
ขึ้นอยู่กับประสบการณ์



```
Form1.vb X Form1.vb [Design]
Button1
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim i As Integer = 2
        Dim total As Int64 = 0
        Do
            total += i
            i += 2
        Loop While i <= 100
        Label1.Text = total
    End Sub
End Class
```

Form1

OK

2550

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : วิธีการประมวลผล algorithm

- จงหาผลรวมของ $1 + 2 + 3 + 4 + 5 + \dots 10000$
- จงหาผลรวมของเลขคู่ $2 + 4 + 6 + \dots 1,000,000$

จงเขียน PSEUDO-CODE รหัสเทียม/รหัสจำลอง
จงเขียนผังงาน (FlowChart)

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : วิธีการประมวลผล algorithm

- จงหาผลรวมของ $1 + 2 + 3 + 4 + 5 + \dots 10000$

START

.... กำหนดค่าเริ่มต้น (Initialization Stage)

.... การรับค่า (INPUT Stage)

... การประมวลผล (PROCESS Stage)

... แสดงผล (OUTPUT Stage)

...

STOP

คำตอบ ไม่สำคัญ
เท่าวิธีการ ?

1

บทที่ 1

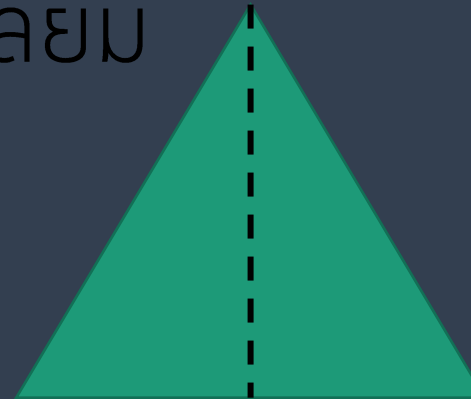
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : วิธีการประมวลผล algorithm

จงคำนวณพื้นที่สามเหลี่ยม

- INPUT Base , Height
- PROCESS สูตร $\frac{1}{2} \times \text{ฐาน} \times \text{สูง}$
- OUTPUT พ.น.



Height

Base

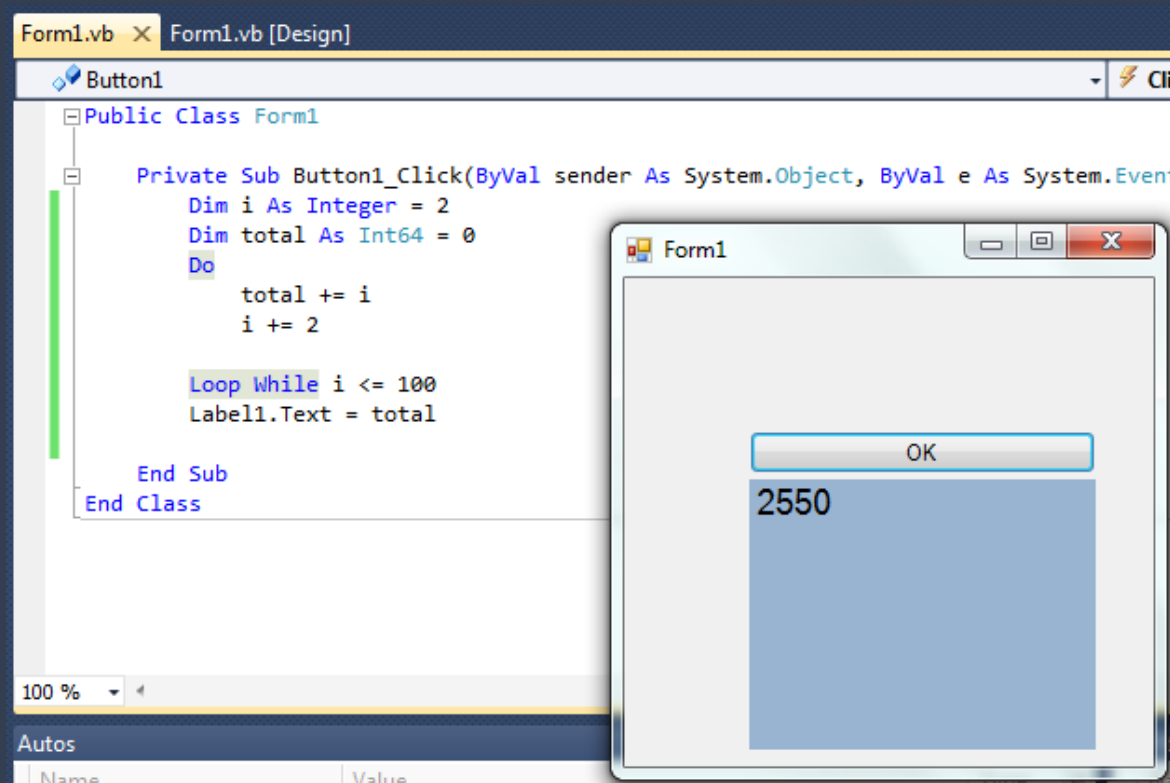
1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : วิธีการประมวลผล algorithm



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : มองว่าเป็นปัญหา แล้วหาวิธีแก้ไขปัญหา

Problem

$$\bullet 2 + 4 + 6 + \dots 100$$

- มีพนักงาน 100 ให้แสดงเงินเดือนมากที่สุด ไปยัง น้อยที่สุด

Max ?

Min ?

Algorithm ขั้นตอนวิธี

Sorting

กำหนดให้มีพนักงาน **100**
ให้แสดงเงินเดือนมากที่สุด ไปยัง น้อยที่สุด

- Algorithm
1. START / BEGIN
 2.
 3. STOP / END / FINISH

INPUT
PROCESS
OUTPUT

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : มองว่าเป็นปัญหา แล้วหาวิธีแก้ไขปัญหา

2 + 4 + 6 + ... 100

- นำเสนอโดย โดยใช้เครื่อง
 - PSEUDOCODE
 - FLOWCHART
- จะต้องอธิบายให้คนอื่น ๆ ในทีมเข้าใจ เพื่อไปพัฒนาต่อ
- เหมือนตอนที่เราร่างบ้าน จำเป็นต้องมีแปลน หรือ ออกแบบ

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

PSEUDOCODE

1. START
2. ... ENTER / READ / INPUT
Number
3. STOP / FINISH / END

1. เริ่มต้น
2. อ่านค่า ตัวเลข 10 จำนวนเข้ามา
3. สิ้นสุด

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

$1 + 3 + 5 + 7 + 9 \dots 999$

- MAX
- MIN
- AVG

8	7	6	8	1
4	9	2	4	25

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

จงเขียน PSEUDOCODE

1. START
2. ENTER Radius
3. COMPUTE CircleArea $\leftarrow$ $3.14 * \text{Radius} * \text{Radius}$
4. Display CircleArea
5. STOP

Assign value

$x \leftarrow 5;$

java

$x = 5;$

if($x == 5$) {

}

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

1. START

1. READ / ENTER >> Initial Stage

2. ประกาศตัวแปร กำหนดค่าเบื้องต้น

3. PROCESS ..การคำนวณ การทำงานเป็นโปรแกรมย่อย

1. ฟังก์ชัน Function

2. โปรซีเยอร์ Procedure

2. แสดงผล Display , Show , Output

Module / Method

3. STOP

printf("hello world");

writeln("hello world");

A = pow(2,3)

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : Pseudocode TH/EN

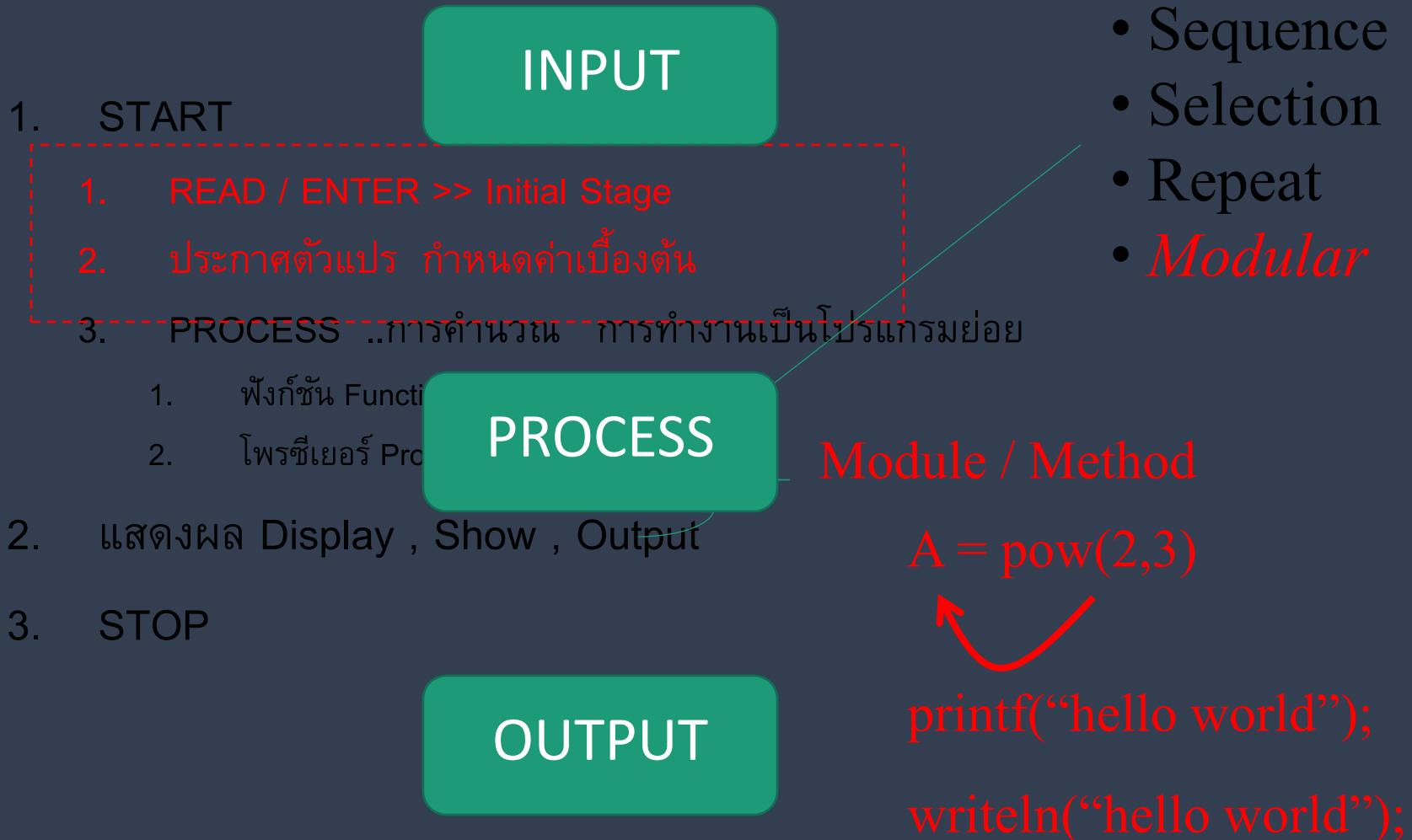
- Algorithm + Syntax

X := 5 ; , pascal , delphi

X = 5; ... java , c

X = 5vb

Pseudocode TH/EN



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

จงเขียนโปรแกรมเรียงลำดับตัวเลข 13 ตัว

• 8 9 7 4 5 1 8 5 3 4 1 2 6

- จงเรียงจากน้อยไปมาก
- จากมากไปน้อย



จงเขียนอัลกอริทึม

- Pseudocode
- FlowChart

จงเขียน Algorithm ของการหาอายุเฉลี่ยของ นศ. ในห้องนี้

ครั้งที่ 1

1. START
2. READ number_Of_student
3. DO / FOR I = 1 TO number_Of_Student
 1. READ age of Student
 2. COMPUTE Total $\leftarrow$ age_Of_student
 3. SET I $\leftarrow$ I + 1
4. COMPUTE AVG $\leftarrow$ Total / number_Of_Student
5. Display "Average Age of Student = " + AVG
6. STOP

จงเขียน Algorithm ของการหาอายุเฉลี่ยของ นศ. ในห้องนี้

ปรับปรุง ครั้งที่ 1

1. START
2. DISPLAY "Plz. Enter Number of Student : "
3. READ *number_Of_student* to NumOfStd
4. REPEAT / DO / FOR I = 1 TO *NumOfStd*
 - 4.1 READ age of Student
 - 4.2 COMPUTE Total $\leftarrow$ age_Of_student
 - 4.3 SET I $\leftarrow$ I + 1
5. COMPUTE AVG $\leftarrow$ Total / *NumOfStd*
6. Display "Average Age of Student = " + AVG
7. STOP

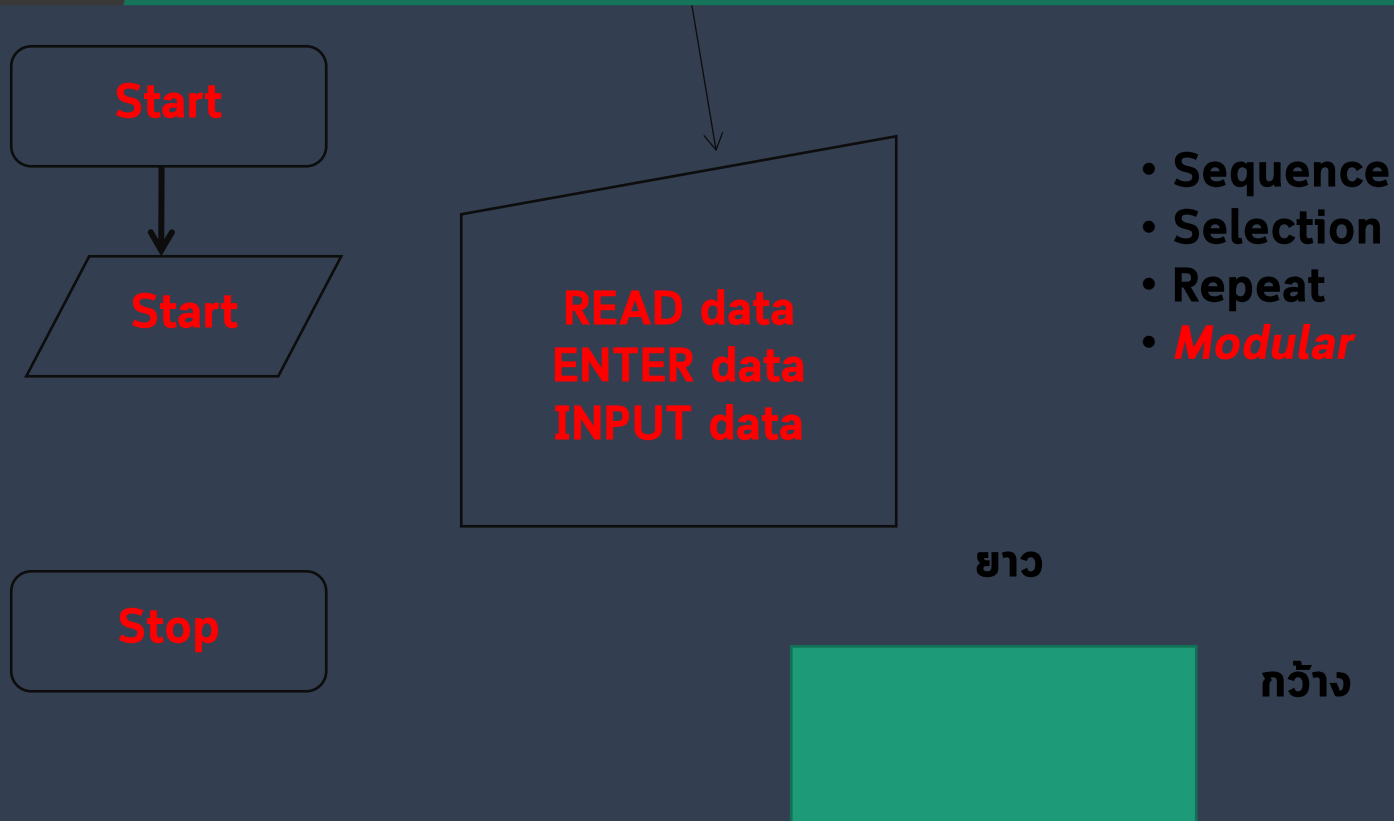
1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

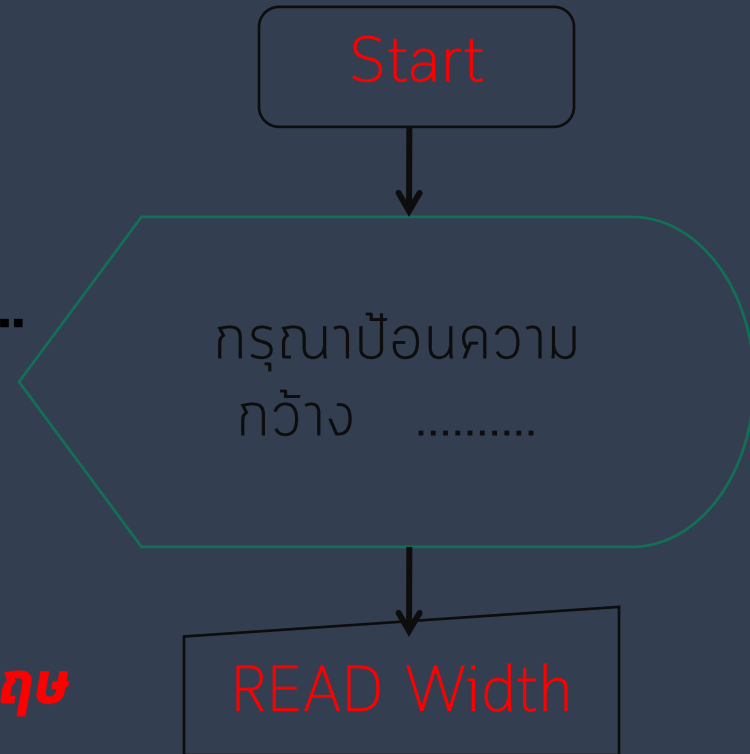
Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

การรับค่า

- กรุณาป้อนความกว้าง
- กรุณาป้อนความสูง

*** ภาษาไทย ไม่ควรปะปนภาษาอังกฤษ



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : รหัสจำลอง (PSEUDOCODE)

จงหาค่าเฉลี่ยของ อายุ นักศึกษาในห้องนี้

- การวิเคราะห์ปัญหา

1. สิ่งที่ต้องพิจารณา
2. รูปแบบผลลัพธ์
3. ข้อมูลนำเข้า
4. ตัวแปรที่ใช้
5. วิธีการประมวลผล --- Psuedocode

Var

- N แทน จำนวน นศ.
- age [10]
- age[N]
- Sum
- Average

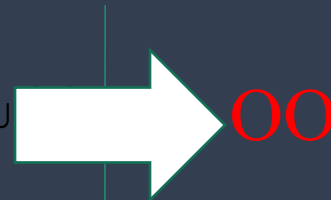
จงหาค่าเกรดเฉลี่ยของ นักศึกษาในห้องนี้

- การวิเคราะห์ปัญหา

- สิ่งที่โจทย์ต้องการ
- รูปแบบผลลัพธ์
- ข้อมูลนำเข้า
- ตัวแปรที่ใช้
- วิธีการประมวลผล --- Psuedocode



หมายถึง แนวทางแบบดั้งเดิม ไม่สามารถนำกลับมาใช้ใหม่ได้
เนื่องจาก การพัฒนาระบบแบบนี้ใช้ได้เฉพาะงาน และไม่ได้คำนึงถึงการนำกลับมา
เนื่องจาก ระบบงานสมัยก่อนไม่ได้ ใหญ่ และมีความซับซ้อนมากนั้น



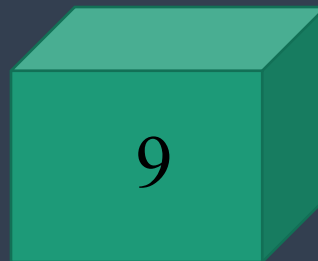
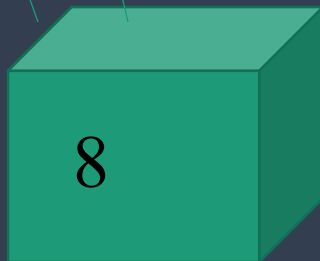
จงเขียนโปรแกรมเรียงลำดับตัวเลข 13 ตัว

8 9 7 4 5 1 8 5 3 4 1 2 6

จงเรียงจากน้อยไปมาก
จากมากไปน้อย



จงเขียนอัลกอริทึม



$N - 1$ รอบ

$N - 1$ รอบ

Max

SORTING

- Selection sort
- Bubble sort
- Merge sort
- Shell sort
- Quick sort
- ۷a۷

ตัวอย่าง : จงแก้ปัญหาคงหา พ.ท. สี่เหลี่ยมใด

1. กว้าง*ยาว
2. จบ



Algorithm ?

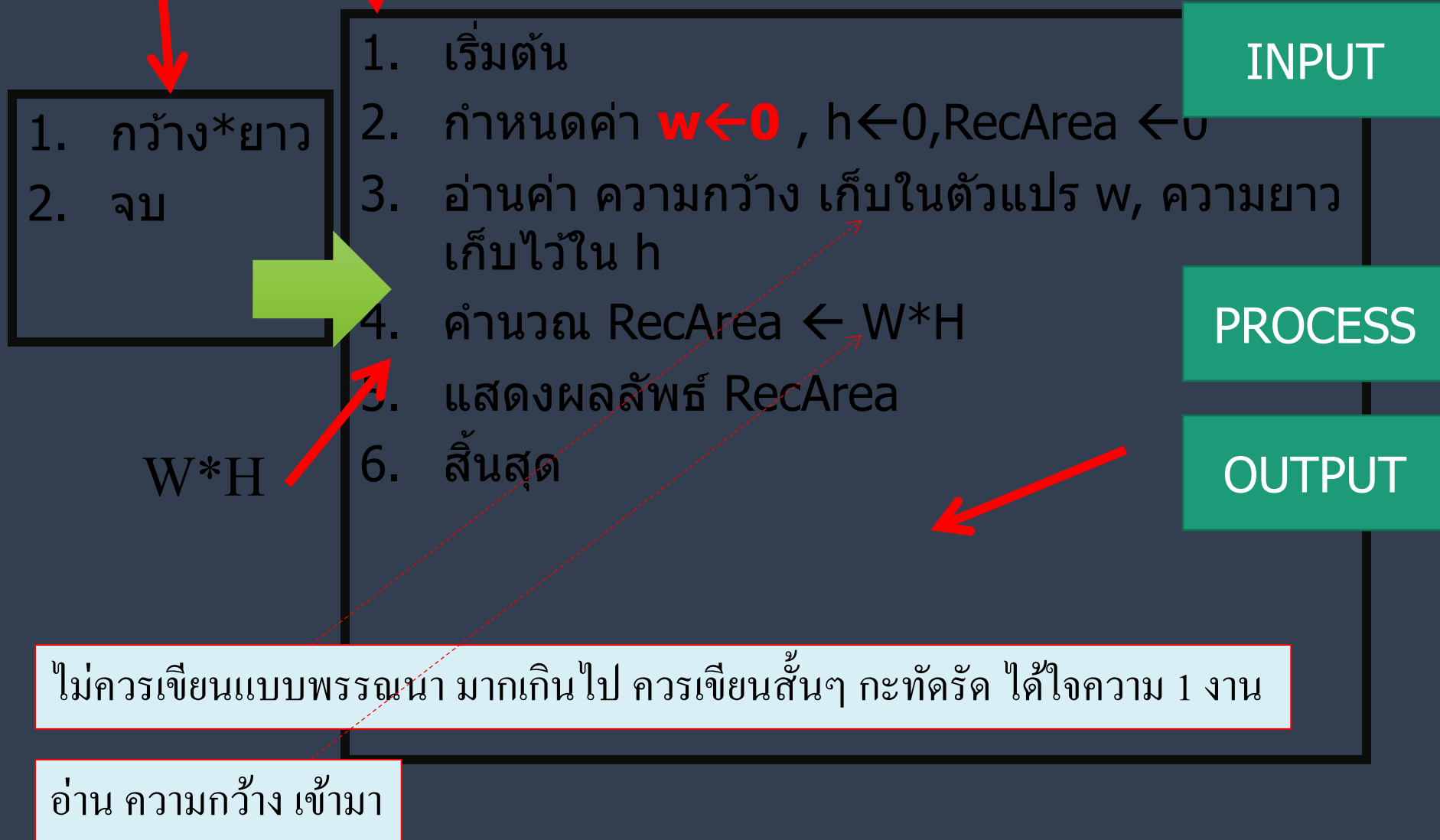
1. $d \times d$
2. จบ



Algorithm ? มีลำดับขั้นตอนที่ชัดเจน

*** การเขียน ออกแบบอัลกอริทึม ต้องทำหลายๆ รอบ

SET $w \leftarrow 0$



PSEUDOCODE

- SET $X \leftarrow 5$;
- SET $X \leftarrow 5$

การกำหนดค่า
Assign value

pascal

$X := 5$;

Operator

=

:=

C/java

$X = 5$;

VB

$X = 5$

C

```
int x;  
x = 10;
```

```
public class calRecArea {  
    public static void main(String args[]) {  
        int width=0;  
        int height=0;  
        int result =0;  
        result =width*height;  
        System.out.println( result);  
    }  
  
}
```

File: calRecArea.java

Case-sensitive

ตัวอย่างใน visual basic

โปรแกรมย่อย ไม่มีการคืนค่า

```
Class CalRecArea
```

```
    Private sub calRecArea()
```

```
        .....
```

```
    End sub
```

```
End class
```

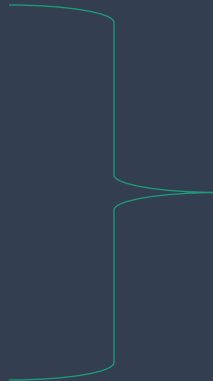
ตัวอย่างใน visual basic

Private function ชื้อข้าว(*money as int16*)

ชื้อข้าว

return ข้าว

End function



Statement ประโยคคำสั่ง

หาตัวเลขยกกำลัง

vb

```
Private function pow(x as int16,y as int16) as  
double
```

```
...
```

```
...
```

```
...
```

```
    return x
```

```
end function
```

จงเขียน/สร้างฟังก์ชัน ด้วย

- Java
- C
- Pascal
- Basic
- VB
- VC#

Case-sensitive

- *Case-sensitive*
 - *java , c ,VC# , c++ ,c# ,php ,perl*
- *None-case sensitive*
 - *Vb , delphi ,pascal*

int x;

int X;

การยกกำลัง (power)

- 2^5
- `Math.pow(2,5)`

*Procedure/sub/void
Function*

java

- int x;
- int X;
- int Salary;
- int salary;

- Vb
- Dim x as int16;
- Dim X as int32;

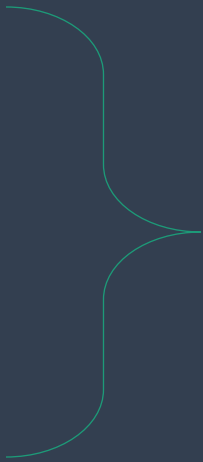
จงเขียน Algorithm ของ

- $1 + 2 + 3 + \dots + 100$

- *Do*

- *While*

- *For*



เลือกใช้อย่างใดอย่างหนึ่ง

นำเสนอ Pseudocode และ Flowchart

ครั้งที่ 1

1 100

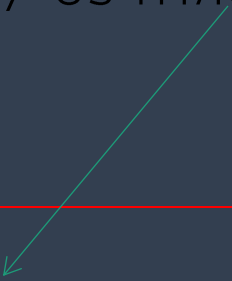
• Algorithm : ขั้นตอนวิธี / การแก้ปัญหาย่อยเป็นขั้นตอน

1. เริ่มต้น
2. กำหนดค่า 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

*** ค่าคงตัว รู้แน่นอน
ไม่รับค่า
ควรเลือก for ,while,do

ปรับปรุงครั้งที่ 1

กำหนดค่า / ประกาศตัวแปร



1. เริ่มต้น
2. ทำตั้งแต่ 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

1. START/BEGIN
2. {*Initialization*}
3. DECLARE result
4. STOP/FINISH/END

ปรับปรุงครั้งที่ 1

กำหนดค่า / ประกาศตัวแปร

1. เริ่มต้น
2. ทำตั้งแต่ 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

1. START/BEGIN
2. {*Initialization*}
3. SET I $\leftarrow$ 0
4. STOP/FINISH/END

Assign value

SET X $\leftarrow$ 8
SET X $\leftarrow$ X + 8

X = 5
X = 5;
X :=5;

vb

C,java

Pascal/delphi

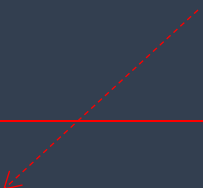
กำหนดค่าให้กับตัวแปร Assign value

- SET SUM $\leftarrow$ SUM + 5
- SET Y $\leftarrow$ 8
- Total $\leftarrow$ Total*25

ปรับปรุงครั้งที่ 2

กำหนดค่า / ประกาศตัวแปร

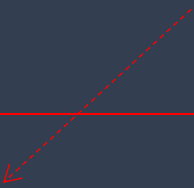
1. เริ่มต้น
2. ทำตั้งแต่ 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

1. START/BEGIN
 2. {*Initialization*}
 3. SET I $\leftarrow$ 0
 4. FOR I $\leftarrow$ 1 TO 100
 - 4.1 BEGIN
COMPUTE RESULT $\leftarrow$ RESULT + I
 - 4.2 END
 5. STOP/FINISH/END
- 

ปรับปรุงครั้งที่ 3

กำหนดค่า / ประกาศตัวแปร

1. เริ่มต้น
2. ทำตั้งแต่ 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

1. START/BEGIN
 2. {*Initialization*}
 3. SET I $\leftarrow$ 0
 4. SET result $\leftarrow$ 0
 5. FOR I $\leftarrow$ 1 TO 100
 - 4.1 BEGIN
 - COMPUTE result $\leftarrow$ result + I
 - 4.2 END
 6. STOP/FINISH/END
- 

ปรับปรุงครั้งที่ 3

กำหนดค่า / ประกาศตัวแปร

1. เริ่มต้น
2. ทำตั้งแต่ 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

1. START/BEGIN
2. {*Initialization*}
3. SET I $\leftarrow$ 0
4. SET total $\leftarrow$ 0
5. FOR I $\leftarrow$ 1 TO 100
 - 4.1 BEGIN
 - COMPUTE total $\leftarrow$ total + I
 - 4.2 END
6. STOP/FINISH/END

Result

Sum

Total

ปรับปรุงครั้งที่ 3

กำหนดค่า / ประกาศตัวแปร

1. เริ่มต้น
2. ทำตั้งแต่ 1 ...100
3. เอาแต่ละค่าบวกกัน
4. แสดงผลลัพธ์
5. สิ้นสุด

1. START/BEGIN
2. {**Initialization**}
3. SET I $\leftarrow$ 0
4. SET total $\leftarrow$ 0
5. FOR I $\leftarrow$ 1 TO 100
 - 4.1 BEGIN
 - COMPUTE total $\leftarrow$ total + I
 - SET I $\leftarrow$ I + 1
 - 4.2 END
6. STOP/FINISH/END

Result
Sum
Total

DISPLAY
SHOW
PRINT
OUTPUT

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : ผังงาน (Flowchart)

ประเภทของผังงาน (Flowchart)

- Program Flowchart : ผังงานโปรแกรม แสดงรายละเอียด
- System Flowchart : ผังงานระบบอ ภาพรวม

1

บทที่ 1

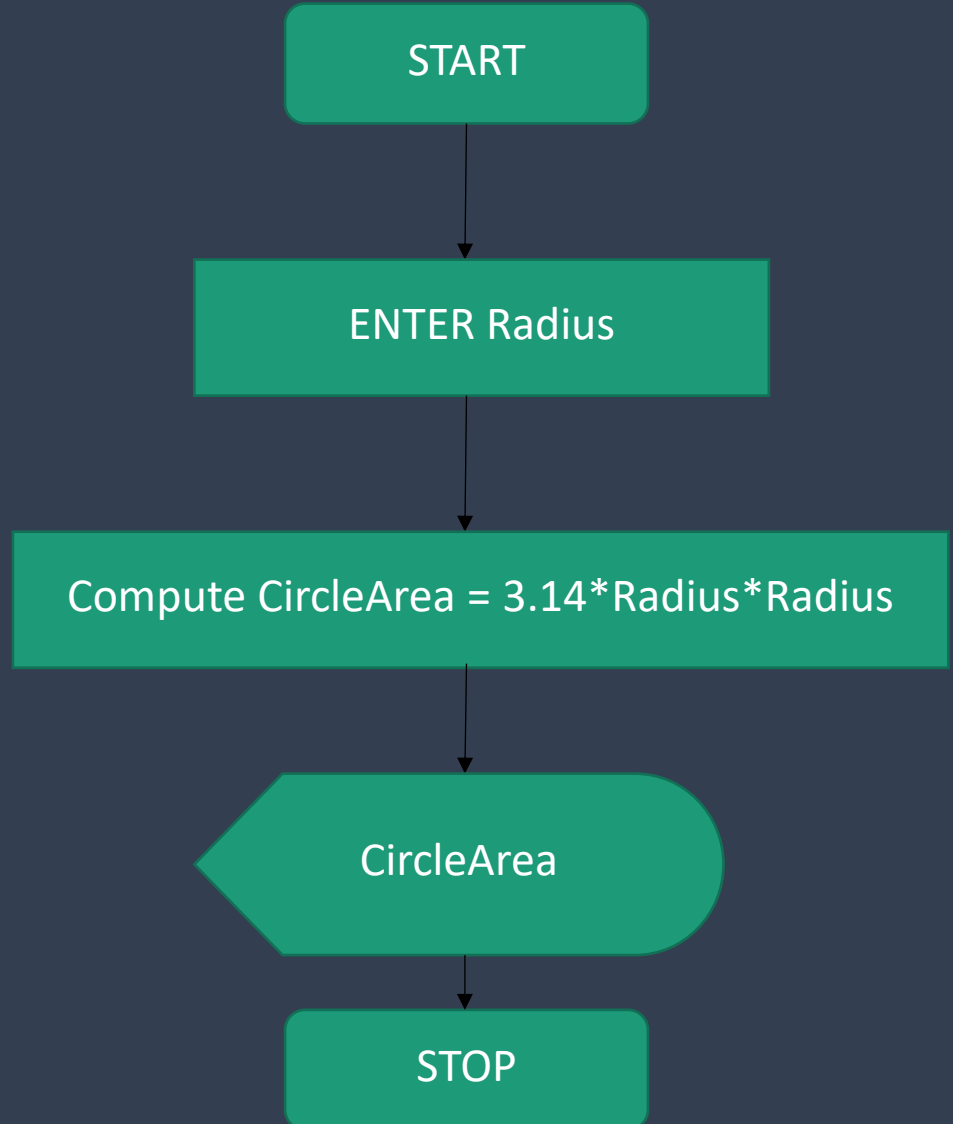
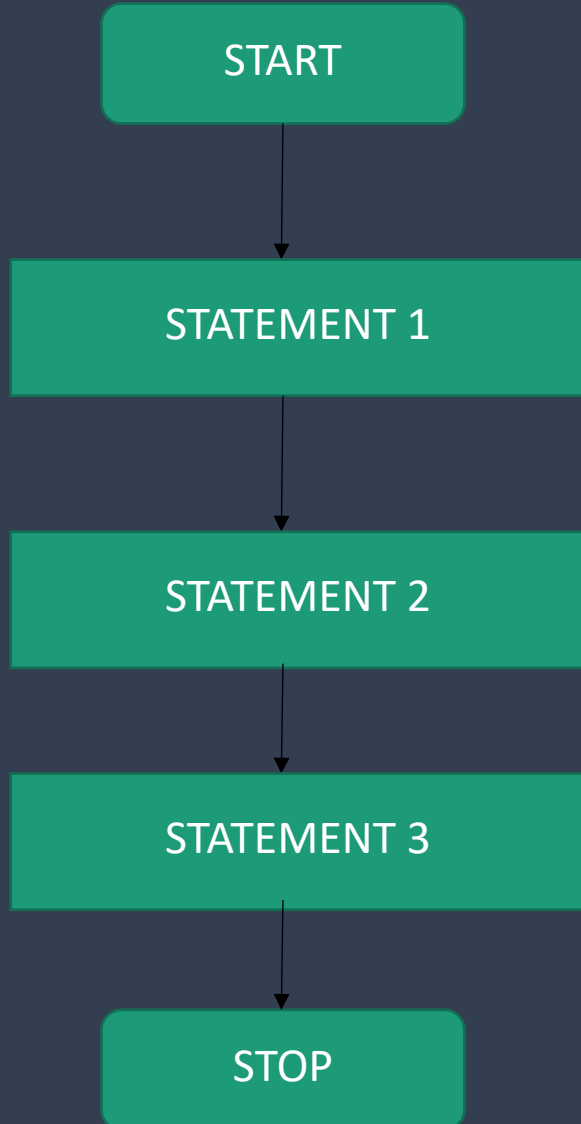
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : ผังงาน (Flowchart)



Sequential



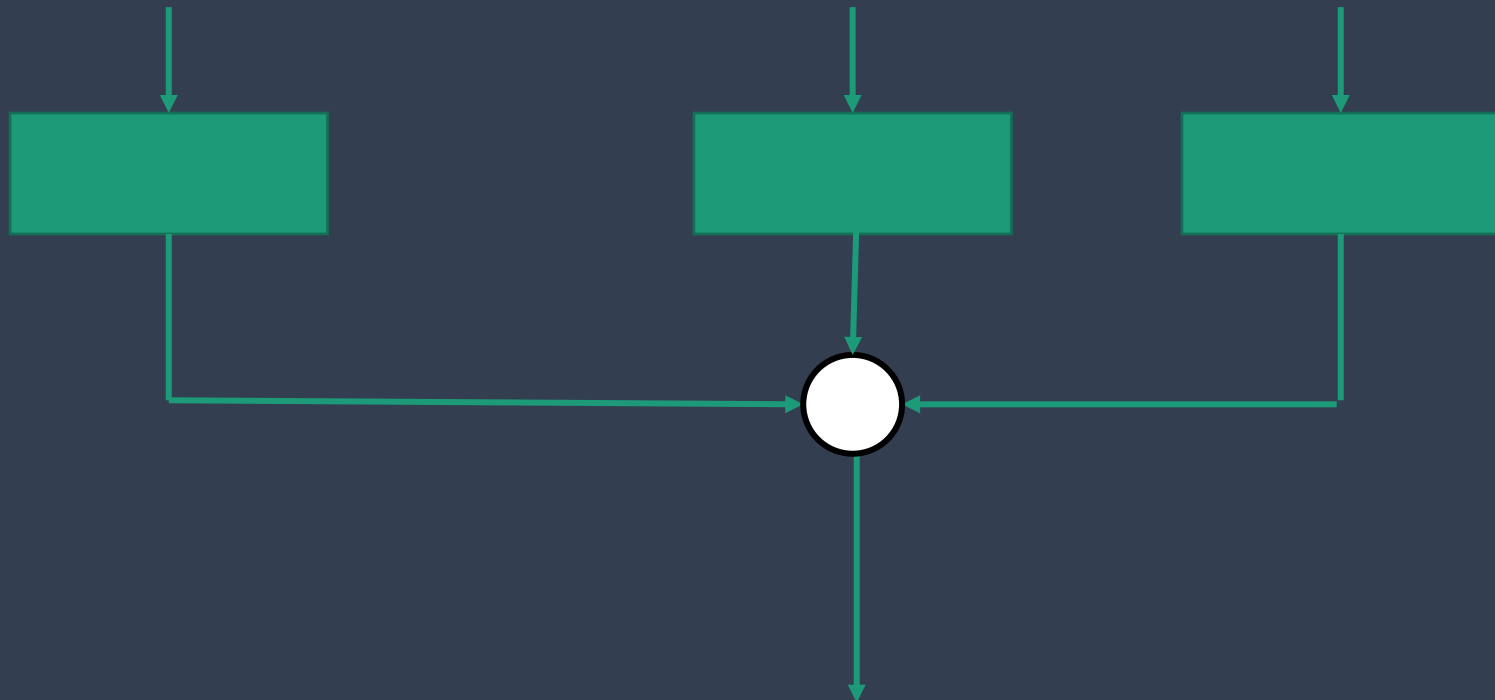
1

บทที่ 1

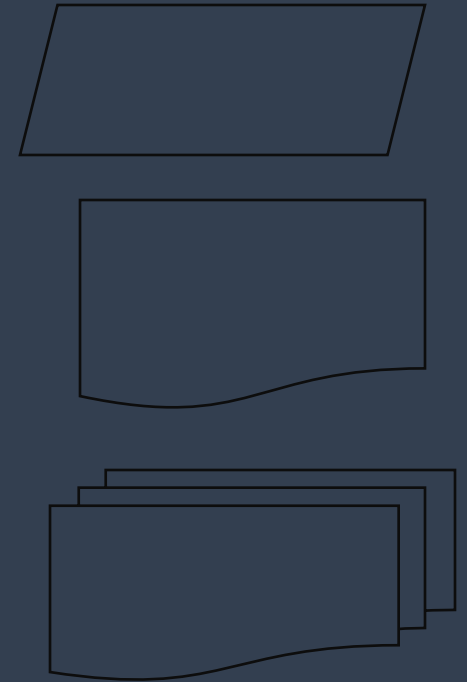
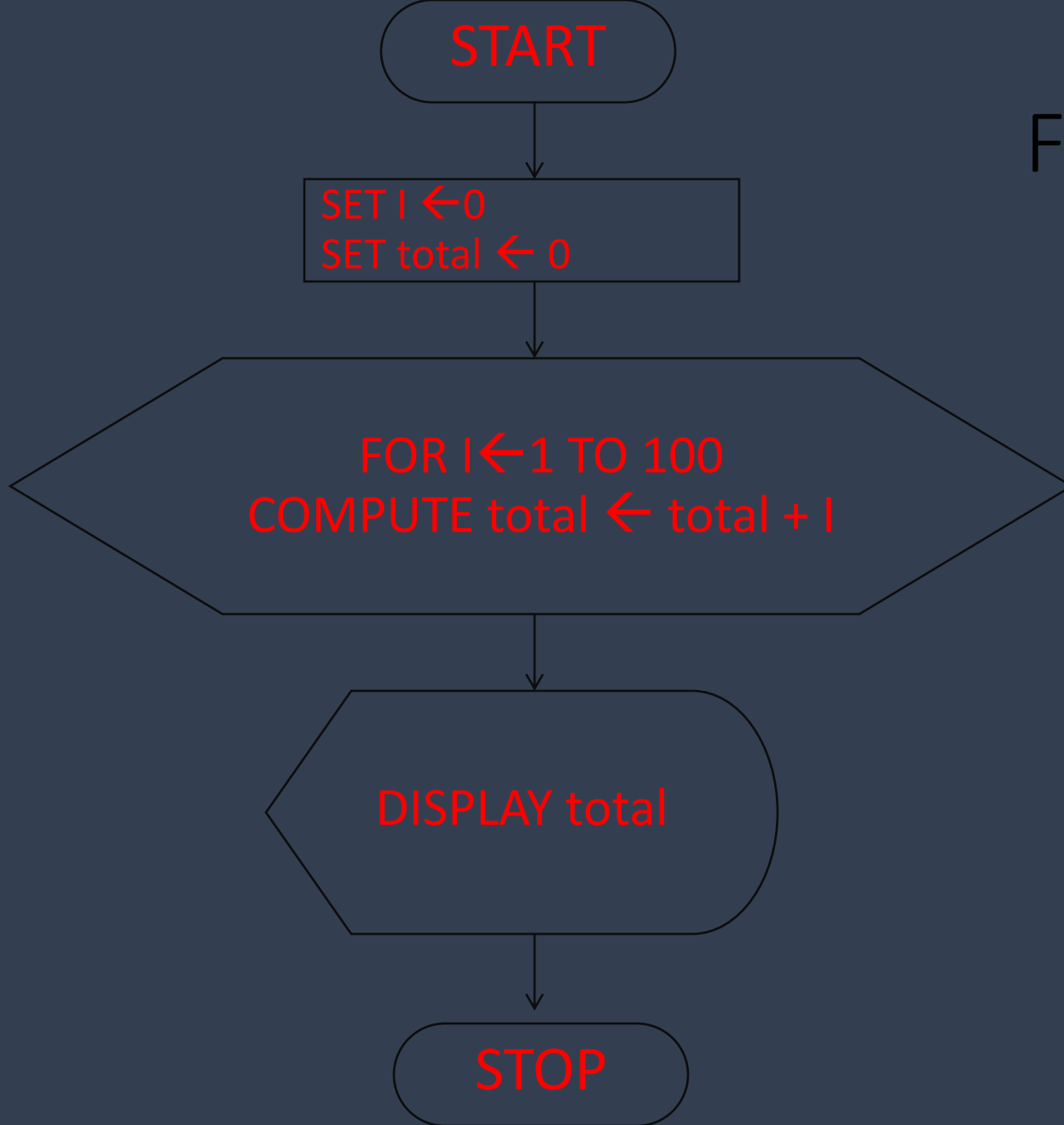
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

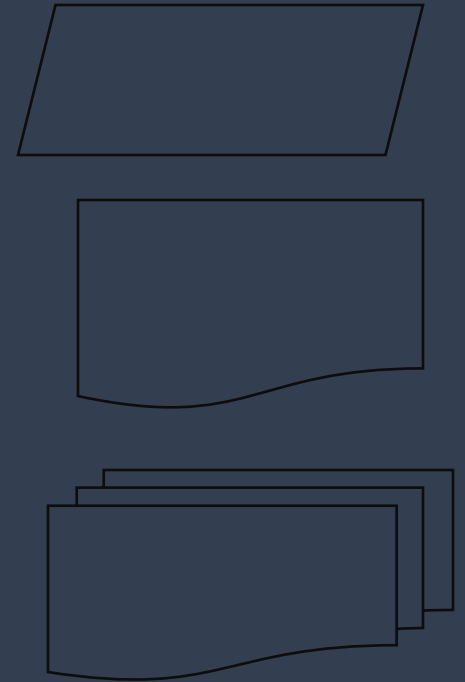
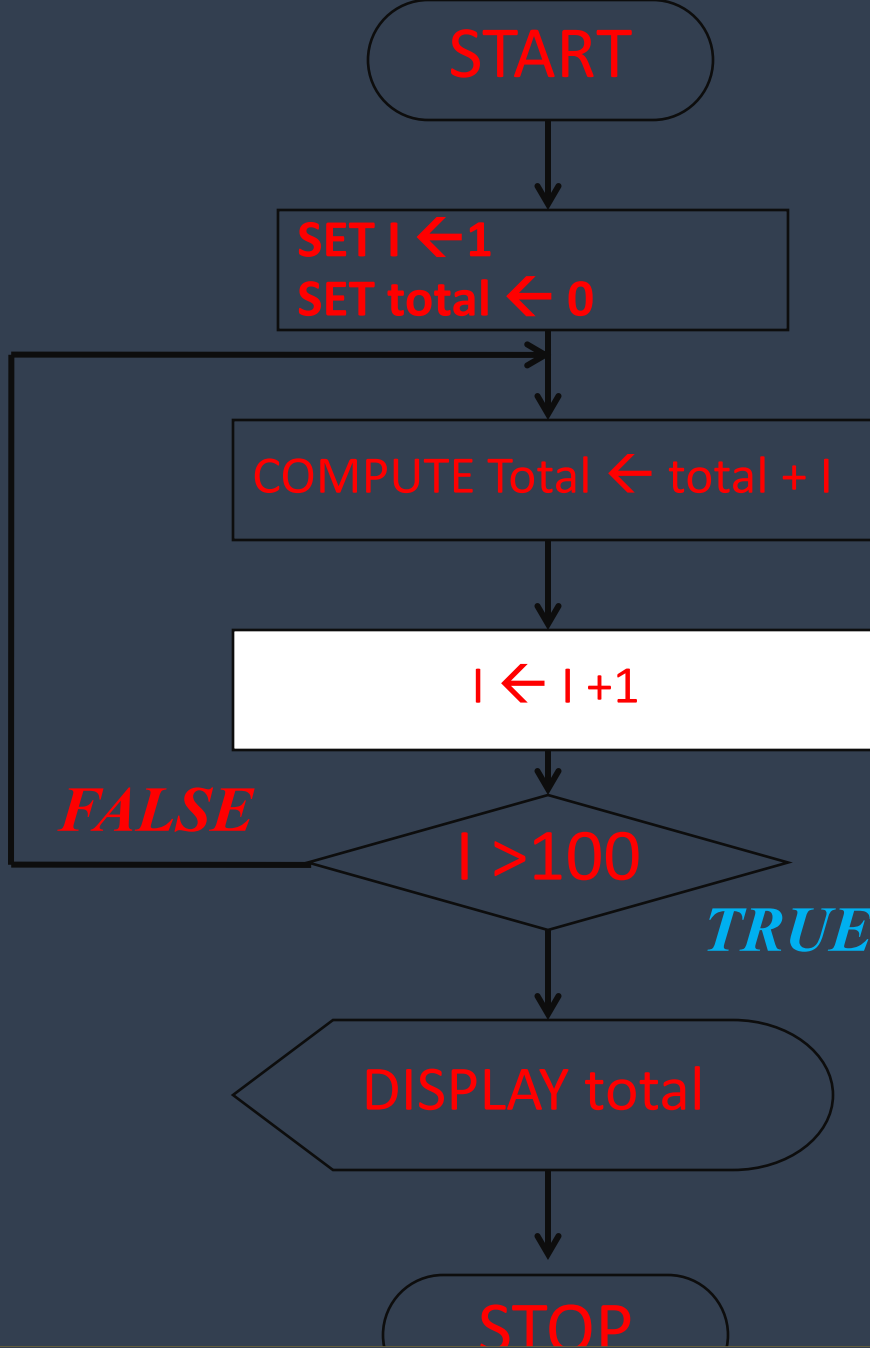
Algorithm : ผังงาน (Flowchart)



FLOWCHART



FLOWCHART DO



1

บทที่ 1

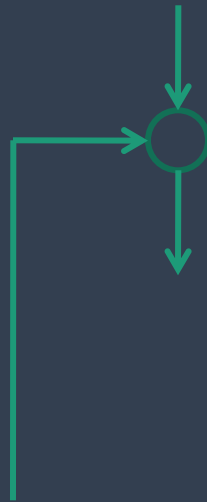
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : ผังงาน (Flowchart)

connector

Same page



Previous page



Next page



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : Desktop Application การพัฒนาโปรแกรม ประยุกต์บนวินโดวส์

- OOP
- Event-Driven Programming
- Visual Programming



VB
Delphi
Java
VC#

IDE
Integrate Development Environment

Visual Studio.NET2019

NetBeanIDE
Android Studio

Tools

Language

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

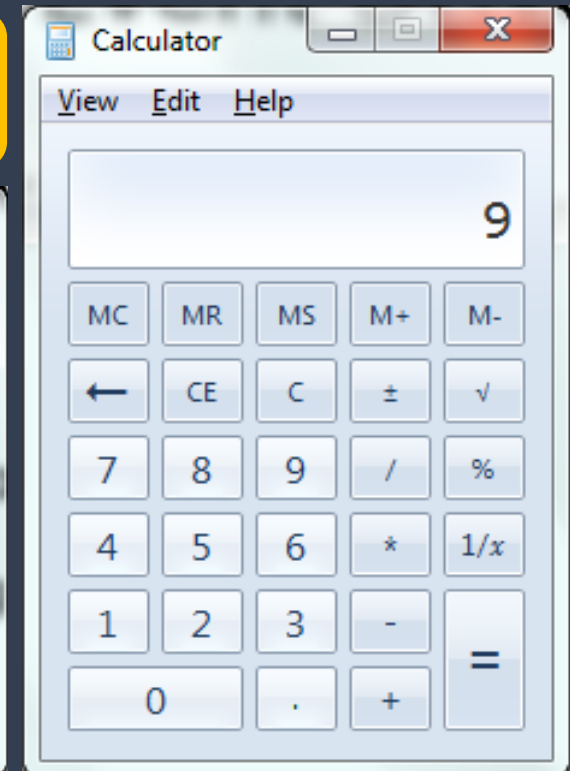
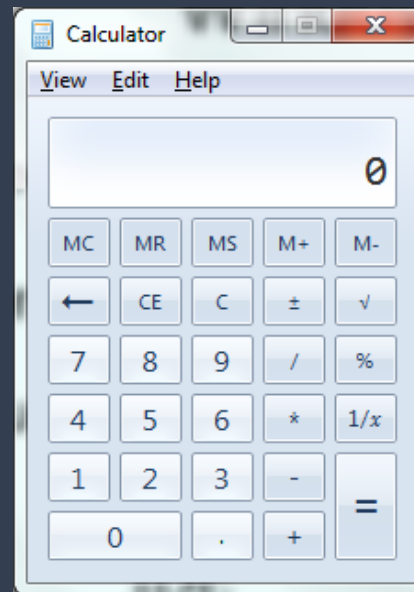
Reviews

Algorithm : Event Driven Programming

การเขียนโปรแกรมตามเหตุการณ์ (Event Driven Programming)

Button9_click()

Button8_click()



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : Event Driven Programming

- การเขียนโปรแกรมแบบอ้างอิงเหตุการณ์
- Keyup
Keydown
keypress



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

Algorithm : Text Mode / GUI

Text Mode / GUI

Program test

Begin

...

...

...

End.

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

แนวทางการพัฒนาซอฟต์แวร์

Traditional/Structural

- Structural Programming (basic, c, pascal)
- Algorithm
- Flowchart
- DFD
- ER
- Normalization

Object-Oriented

- Unified Process
- UML
 - use case diagram
 - class/object diagram
 - Activity diagram
 - Sequence diagram
 - Collaboration diagram
 - deployment diagram
 - component diagram

Requirement Specification



Information/Database System

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

แนวทางการพัฒนาซอฟต์แวร์ : Structural Programming

Structural Programming

- C , Pascal
- Basic >> none Structural Programming / none Procedural Programming
 - GOTO 28
 - GOTO 52
 - GOTO 14

DO , While , For

Sequence / ลำดับ

Selection / Decision ตัดสินใจ ทางเลือก

Iteration / Repeat / Loop การทำซ้ำ

1

บทที่ 1

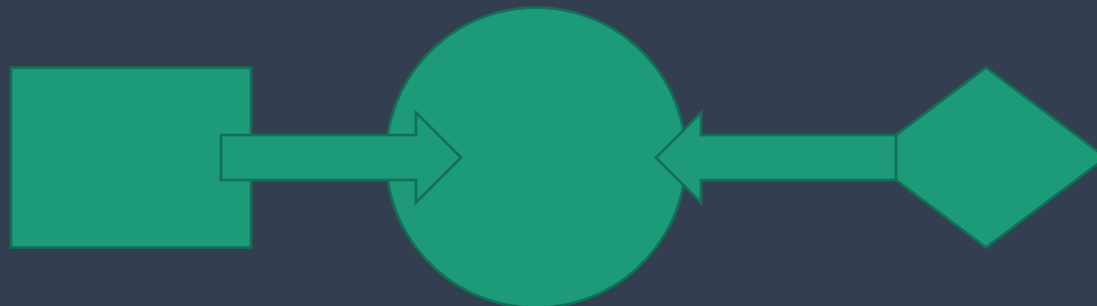
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

แนวทางการพัฒนาซอฟต์แวร์ : Object-Oriented Programming

Reusable การนำกลับมาใช้

- ต.ย. การพัฒนาระบบบัญชี ด้วยภาษาปาสคาล สำหรับ บ. A
- ซอฟต์แวร์นี้ จะใช้ได้สำหรับ บ. นี้เท่านั้น หรือ ใช้ได้เฉพาะงานนี้เท่านั้น
- การนำกลับมาใช้ใหม่ ไม่ได้หมายถึง COPY & PASTE



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

แนวทางการพัฒนาซอฟต์แวร์ : Object-Oriented Programming

โปรแกรม สามารถคำนวณที่ ได้



โปรแกรม สามารถคำนวณที่ ได้ สี่เหลี่ยม อีกรูปแบบหนึ่ง



โปรแกรม สามารถคำนวณที่ ได้ สามเหลี่ยม อีกรูปแบบหนึ่ง

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

Reviews

ระบบสารสนเทศ : Data / Information

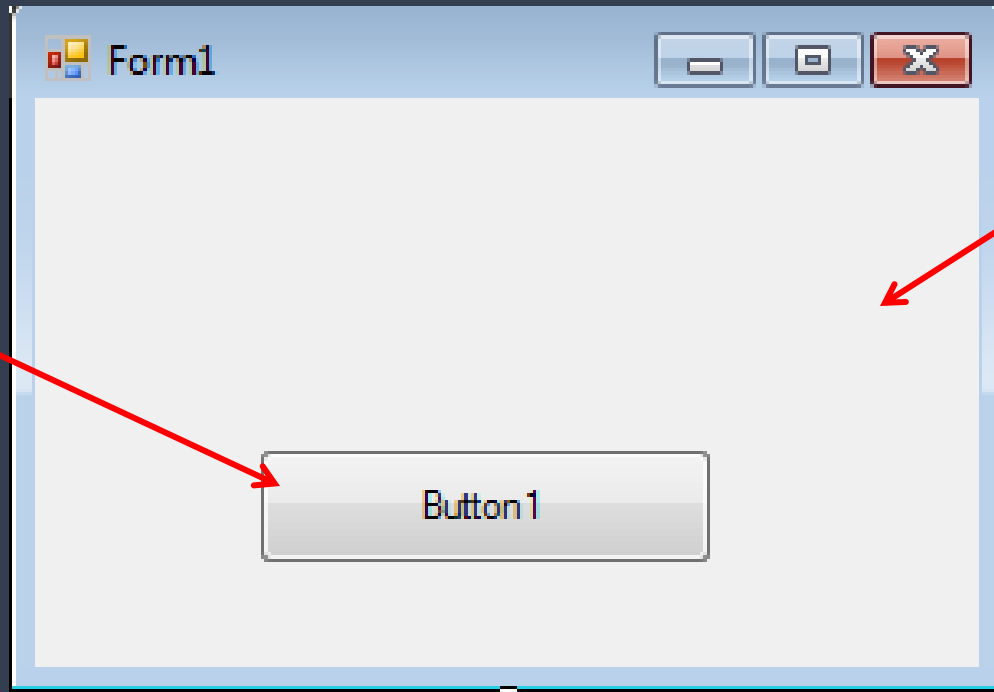
• 8000 , 7999 , 25000 , 7444, 97700, 55000 44400 1000 , 2000 ,

สรุปรายงานแสดงเงินเดือนของพนักงาน

ลำดับ	ชื่อ-นามสกุล	เงินเดือน	โอที
1	นายฐานภัทร	200	0
2	นางสาวรรดา	20000000	150

Summations

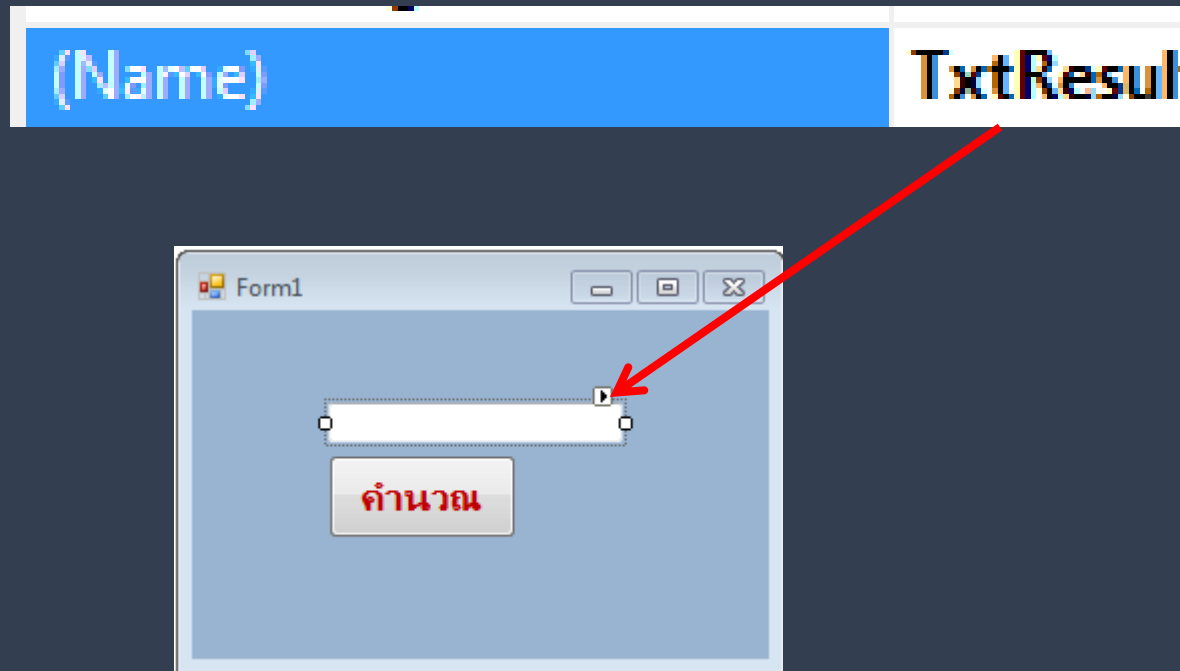
ᄇᄇᄇ Object
- attribute
- function

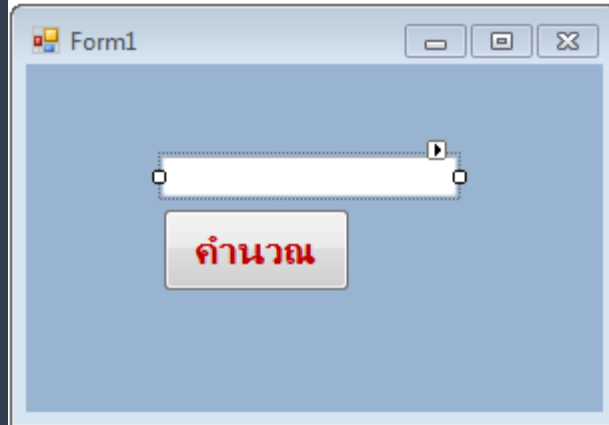


ᄇᄇᄇ Object
- attributes
- functions

TxtResult

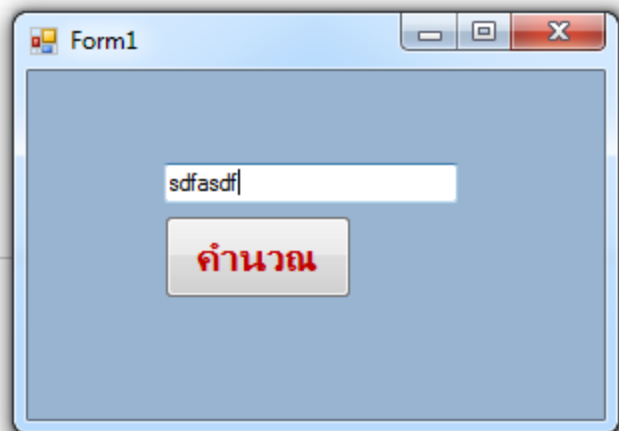
- Form
- Button
- TextBox





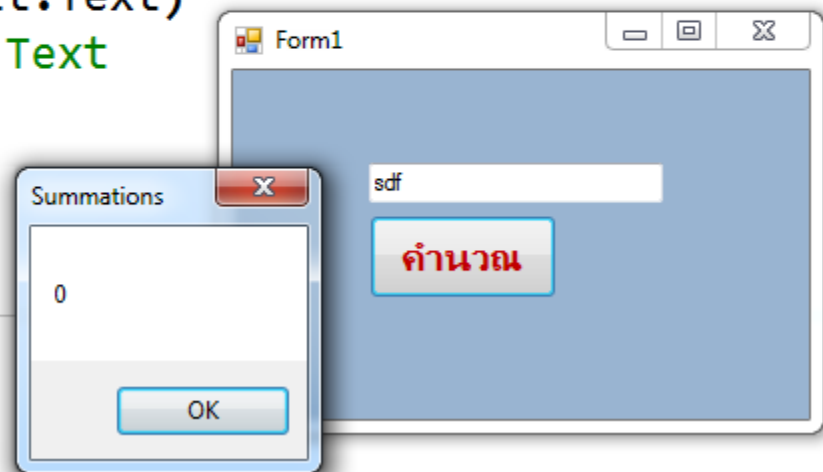
```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnOK.Click
4         TxtResult.Text = 8
5     End Sub
6 End Class
7
```

```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal sender As System.Object)
4         Dim n As Integer
5         'n = Val(TxtResult.Text)
6         n = TxtResult.Text
7
8
9     End Sub
10 End Class
11
```

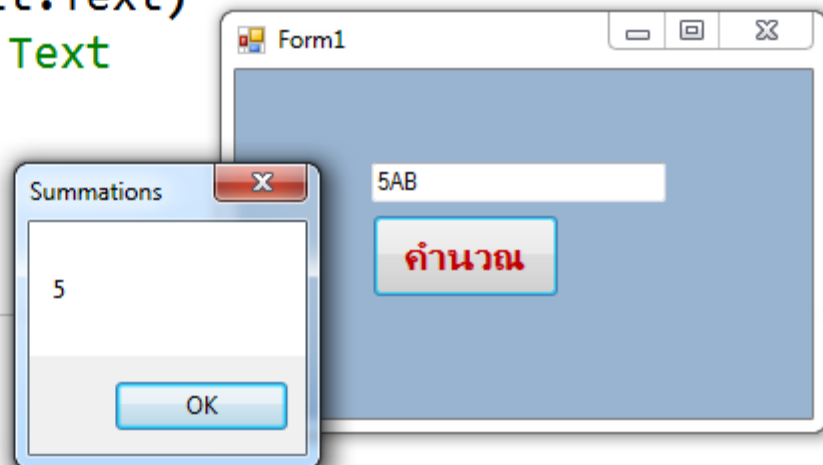


```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal
4         Dim n As Int16
5         n = Val(TxtResult.Text)
6         ' n = TxtResult.Text
7         MsgBox(n)
8
9     End Sub
10 End Class
11
```

```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal sender As System.Object
4         Dim n As Int16
5         n = Val(TxtResult.Text)
6         ' n = TxtResult.Text
7         MsgBox(n)
8
9     End Sub
10 End Class
11
```



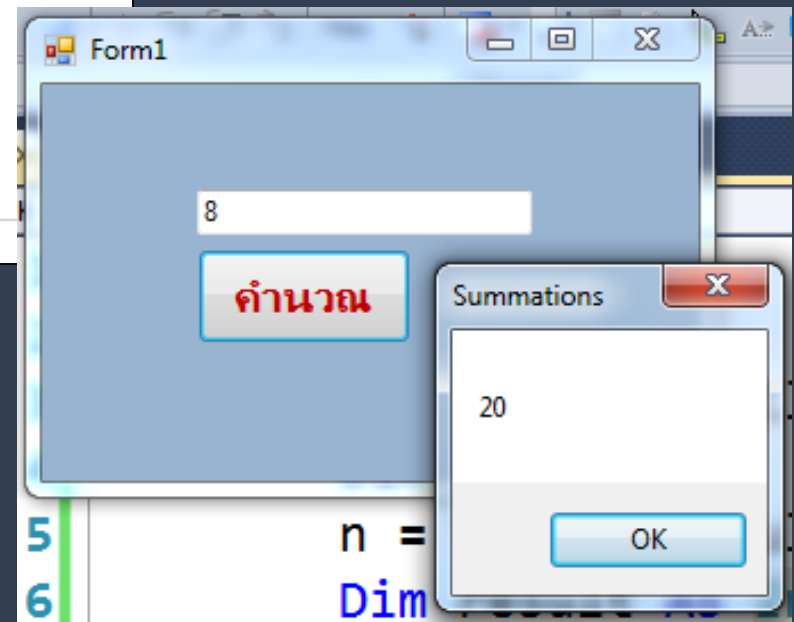
```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal sender As System.Object
4         Dim n As Int16
5         n = Val(TxtResult.Text)
6         ' n = TxtResult.Text
7         MsgBox(n)
8
9     End Sub
10 End Class
11
```



```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal
4         Dim n As Int16
5         n = Val(TxtResult.Text)
6         Dim result As Int32
7
8         For i As Int16 = 1 To n
9             result = result + i
10        Next
11        MsgBox(result)
12    End Sub
13 End Class
```

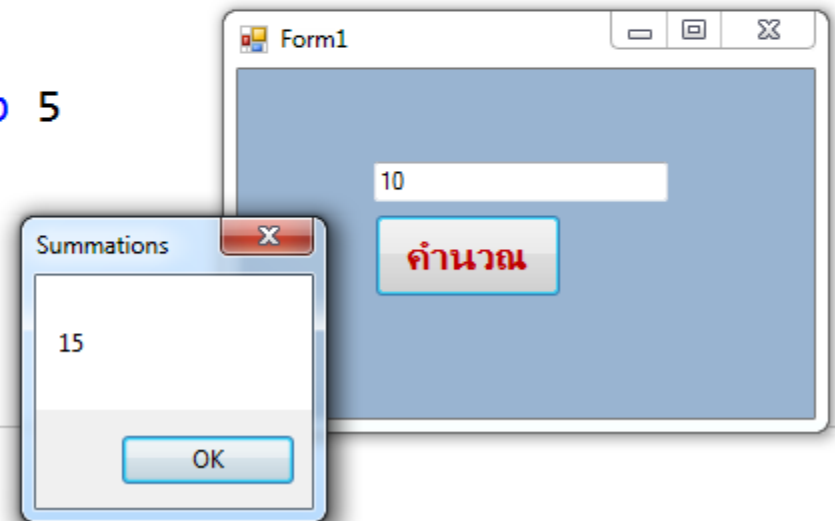
$$2 + 4 + 6 + 8 \dots 100$$

```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal sender
4         Dim n As Int16
5         n = Val(TxtResult.Text)
6         Dim result As Int32
7
8         For i As Int16 = 2 To n Step 2
9             result = result + i
10        Next
11        MsgBox(result)
12    End Sub
13 End Class
14
```



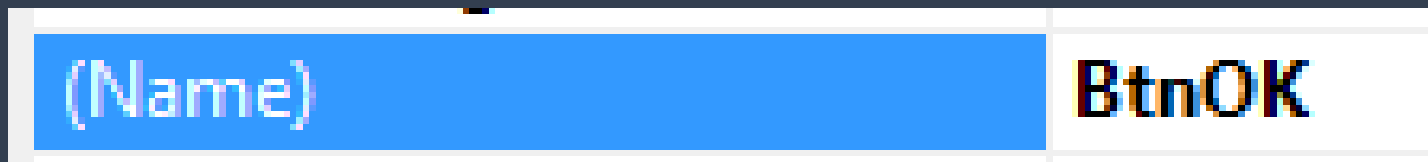
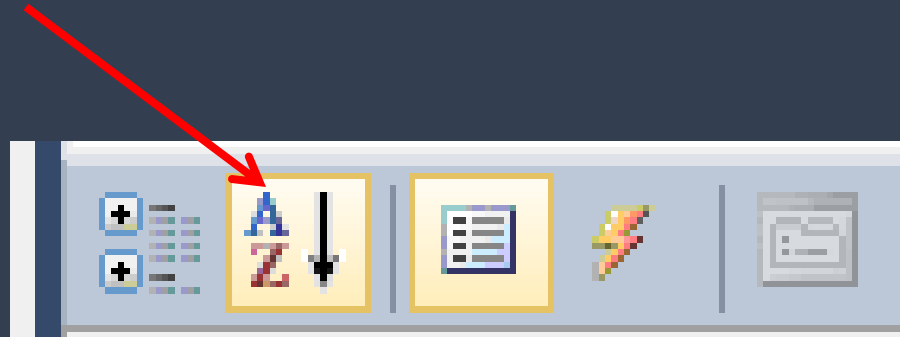
$$5 + 10 + 15 + 20 + \dots 200$$

```
1 Public Class Form1
2
3     Private Sub BtnOK_Click(ByVal sender As System.Object, ByVal e As EventArgs)
4         Dim n As Int16
5         n = Val(TxtResult.Text)
6         Dim result As Int32
7
8         For i As Int16 = 5 To n Step 5
9             result = result + i
10        Next
11        MsgBox(result)
12    End Sub
13 End Class
14
```



Buttons

- Attribute
 - Name
 - Font
 - ForeColor
 - Text
 - BackColor
 - size



1. เริ่มต้น
2. กำหนดค่า $w \leftarrow 0$, $h \leftarrow 0$, RecArea $\leftarrow 0$
3. อ่านค่า ความกว้าง เก็บในตัวแปร w, ความยาว เก็บไว้ใน h
4. คำนวณ RecArea $\leftarrow W * H$
5. แสดงผลลัพธ์ RecArea
6. สิ้นสุด

INPUT

Initialization stage

เตรียมตัวแปร ประกาศตัวแปร

INPUT/READ/ENTER

PROCESS

OUTPUT

PSEUDOCODE : รหัสเทียม/รหัสจำลอง

- START / BEGIN
- SET $X \leftarrow 0$
- READ / ENTER / INPUT
-
- STOP /FINISH / END



- VB
- PASCAL
- C
- JAVA
- DELPHI

← คือ เครื่องหมาย กำหนดค่า assign value

Pascal/Delphi

:=

ในภาษา

X := 5;

Vb/C#/Java

=

ในภาษา

X = 5

X=5;

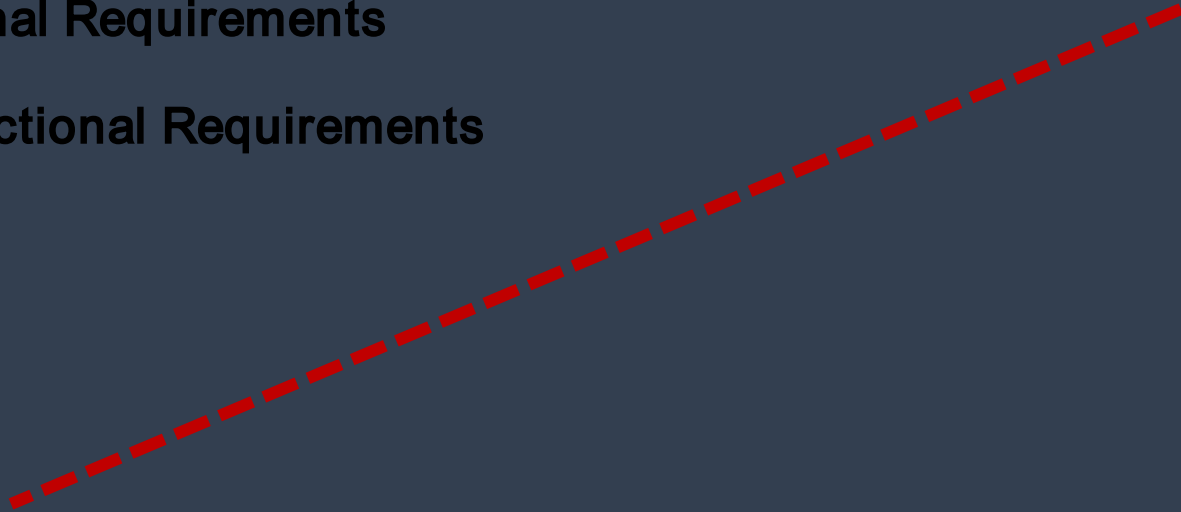


ใน Algorithm

SET $X \leftarrow 5$

Requirement / Information Gathering

- Functional Requirements
- Non-functional Requirements



ภาษาคอมพิวเตอร์/ไม่ใช่ภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์

variables
If /select/switch/
Loop for do while
Function/sub/procedure/method

Compiler/Interpreter

- Pascal
- Vb
- Delphi
- Java
- PHP
- Python
- Etc.

ไม่ใช่ ภาษาคอมพิวเตอร์



สคริป / แท็ก

- HTML
- ASP

+ java script
+ vb / vc#

เทคโนโลยีฝั่งเซิร์ฟเวอร์

Information System :IS

ประเภทระบบสารสนเทศ

- TPS : Transaction Processing System
 - ระบบประมวลผลรายการ เช่น ระบบฝาก-ถอนเงิน ระบบร้านสะดวกซื้อ ระบบเงินเดือน ระบบรับสมัครนักศึกษา
- MIS : Management Information System
 - ระบบทะเบียนนักศึกษา สรุปยอด มีที่คน ลงทะเบียน
 - ระบบบริหารงานบุคคล
 - ระบบคลังสินค้า
- EIS : Executive Information System
 - Graph , Chart
- DSS : Decision Support System
- ES : Expert System (ML, AI)

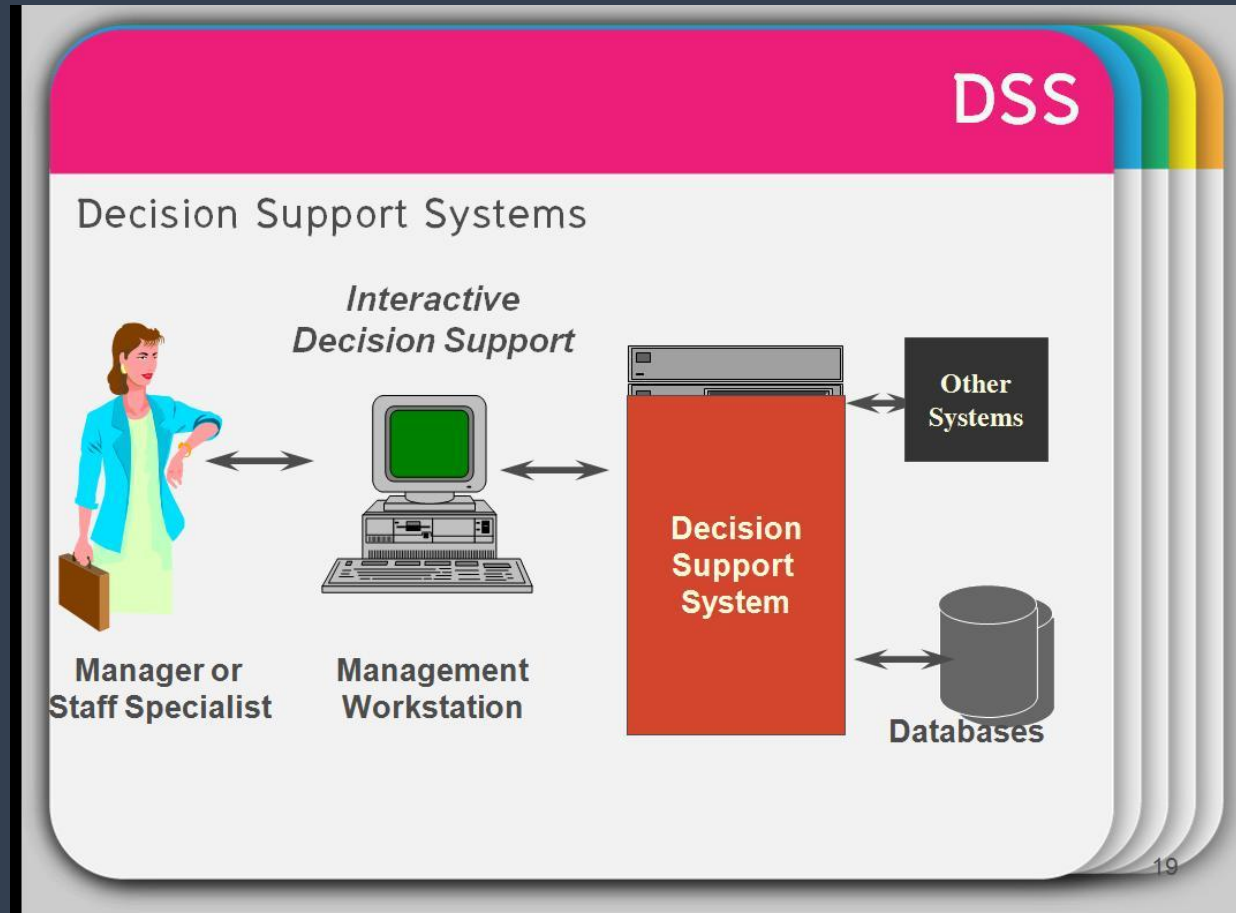
Information System :IS

ประเภทระบบสารสนเทศ

- DSS : Decision Support System
 - คณิตศาสตร์ และ สถิติ มาช่วย ทำให้ง่ายต่อการตัดสินใจ
 - เช่น การตัดสินใจเลือกขยายผลิตภัณฑ์ของโรงงาน
- ES : Expert System ระบบผู้เชี่ยวชาญ
 - ช่วยการตัดสินใจแทนคน
 - เช่น ระบบผู้เชี่ยวชาญการวินิจฉัย
 - ระบบพยากรณ์อากาศ
 - A.I.

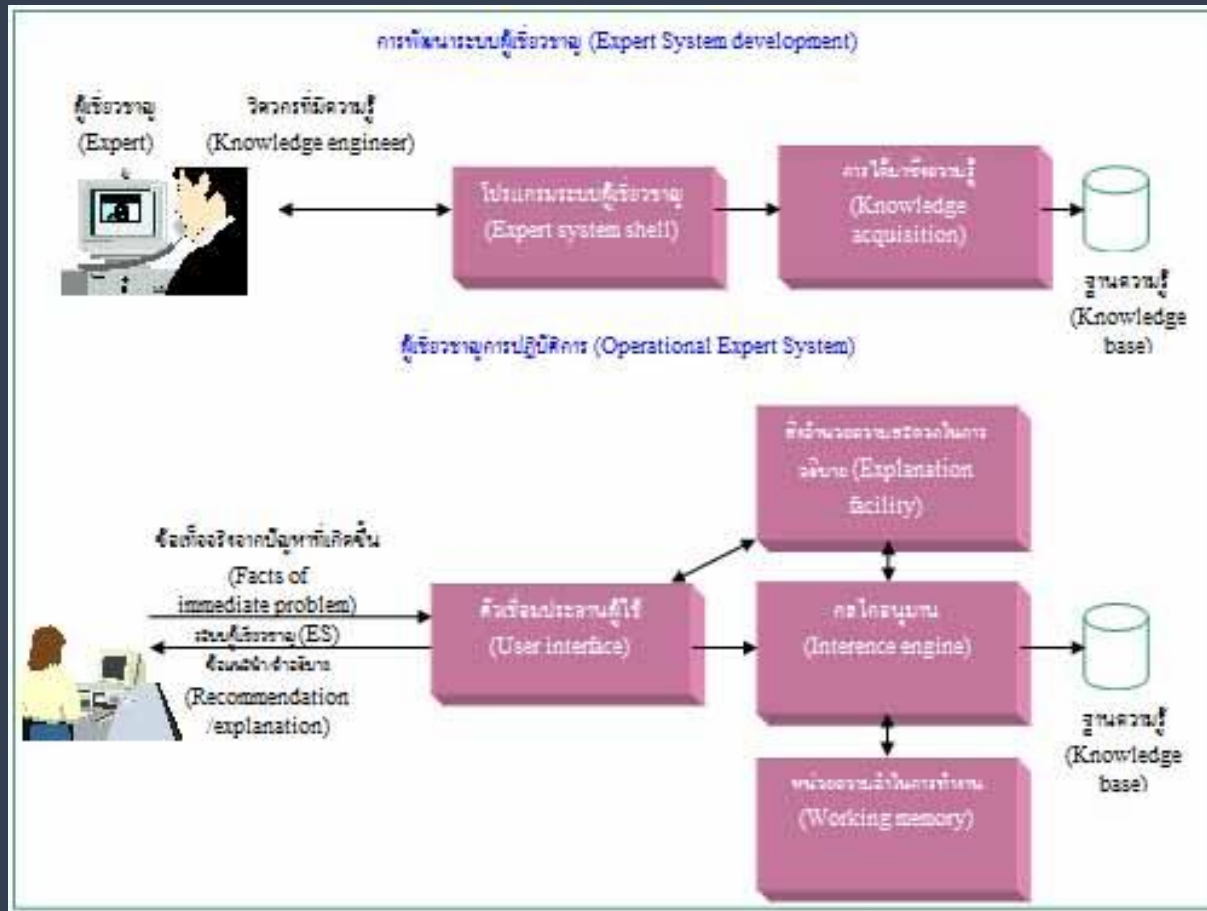
Information System :IS

ประเภทระบบสารสนเทศ



Information System :IS

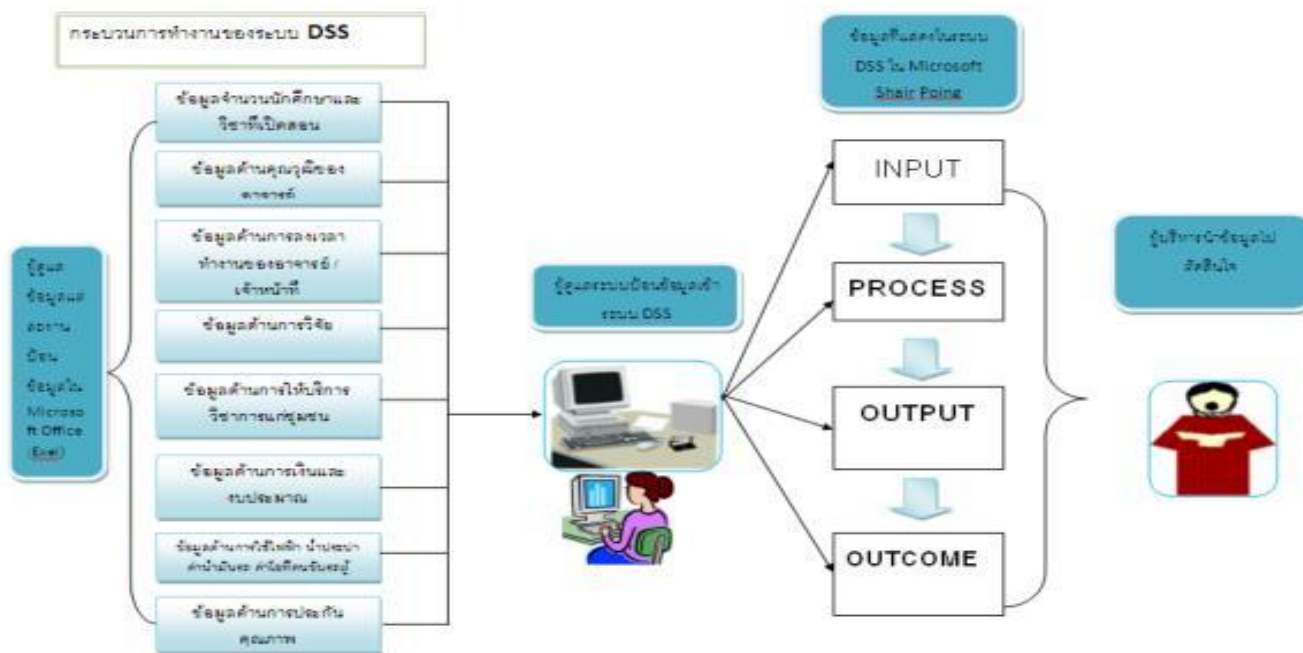
ประเภทระบบสารสนเทศ



Information System :IS

ประเภทระบบสารสนเทศ

กระบวนการทำงานของระบบ DSS



MIS : Inventory

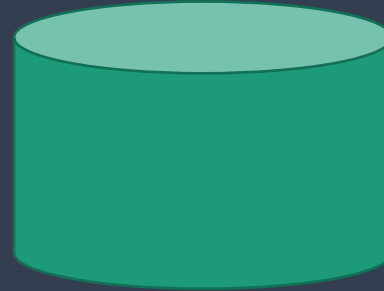
- คน
- สินค้า
- คลังสินค้า
- ตัวสแกนบาร์โค้ด Barcode , RFID ,
- เครือข่าย
- คอมพิวเตอร์
- เครื่องพิมพ์

ประเภทซอฟต์แวร์

- System Software ซอฟต์แวร์ระบบ เช่น windows , dos , Linux
- Application Software
 - Word Processing Ms-word ,
 - Spread Sheet ตารางคำนวณ MS-EXCEL
 - Presentation เช่น Power Point
 - Text Editor ...Notepad , Edit Plus
 - Utility Scanvirus , zip
 - IS ...ระบบสารสนเทศ

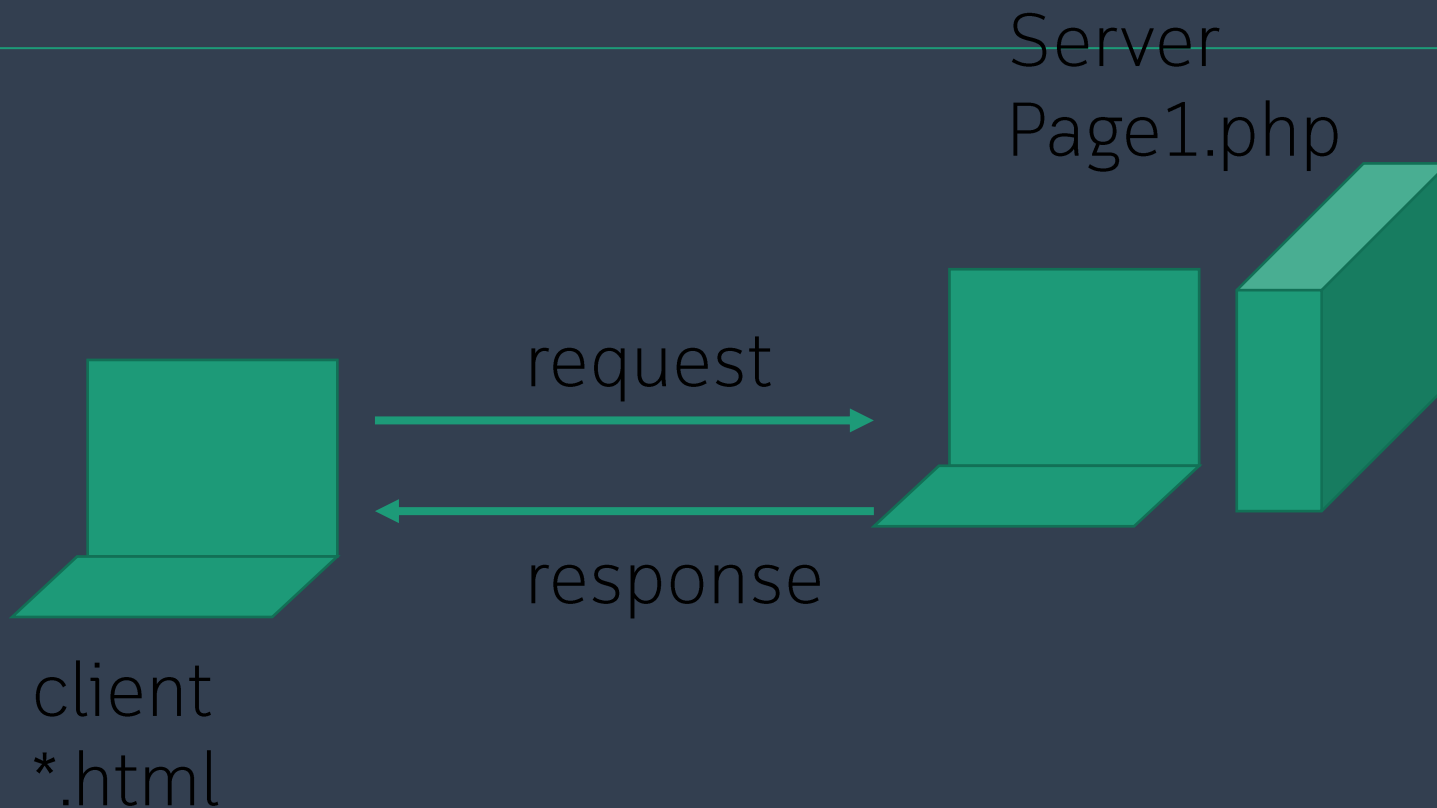
MVC Model

- Model
- View
- Control



Sever Side Technology

- ASP , ASP.NET
- PHP
- JSP



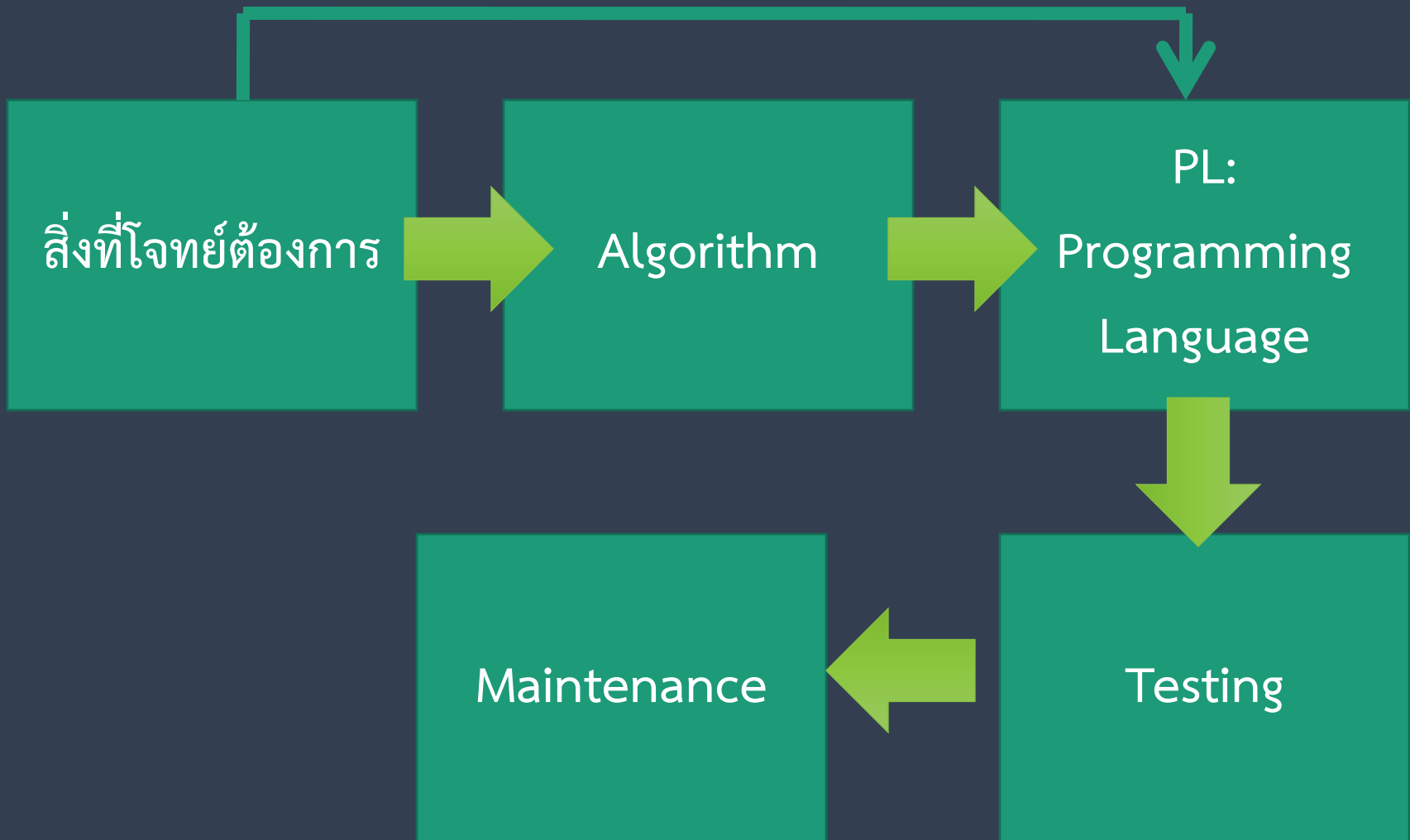
จงเขียน Algorithm ของการหาอายุเฉลี่ยของ นศ. ในห้องนี้

1.

2.

ขั้นตอนการพัฒนาโปรแกรม

SDLC : software development life cycle



สรุปคำศัพท์

- Algorithm
- Traditional Software Development
- Case sensitive , None case-sensitive
- Programming Language
- Divide and Conquer
- Structural Programming
- Traditional Programming
- Server side technology
- IS ระบบสารสนเทศ

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

แนวทางการพัฒนาซอฟต์แวร์ : Object-Oriented Programming

Overview of Object-Oriented Technology

Basic Concepts of Object-Oriented Software Development

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

Agenda

- Software, Systems and Software Systems
- Software Crisis
- Traditional Software Development
- Object-Oriented Software Development
- Principle of Object Orientation
- Key Benefits of Object-Oriented Software Development
- What do we learn in this course?

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

What is Software ?

อะไรคือซอฟต์แวร์ (What is software)

ซอฟต์แวร์ชุดคำสั่ง (Instructions) ที่สั่งให้คอมพิวเตอร์ประมวลผลโดยสามารถทำงานตามที่ระบุไว้ให้สำเร็จ และมีประสิทธิภาพที่ต้องการได้ (computer programs)

โครงสร้างข้อมูล (Data structures) ทำหน้าที่เก็บสารสนเทศที่ใช้ หรือถูกเรียกใช้โดยโปรแกรม

เอกสาร (Documents) ที่บรรยายวิธีปฏิบัติการ และวิธีการใช้งานโปรแกรม (คู่มือ)

- User Manual
- Programmer Manual

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

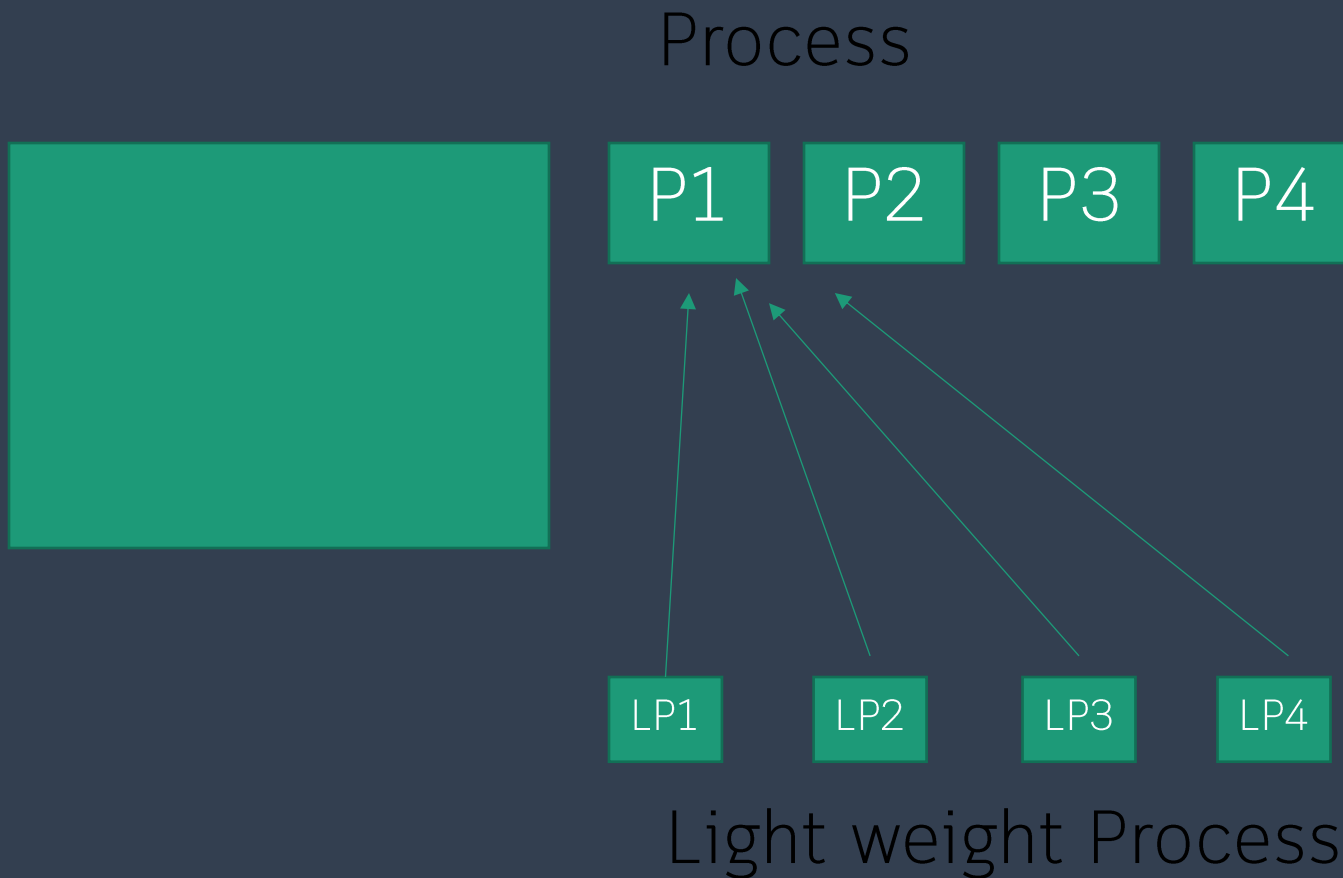
What is Software ?

การทำงานของคอมพิวเตอร์

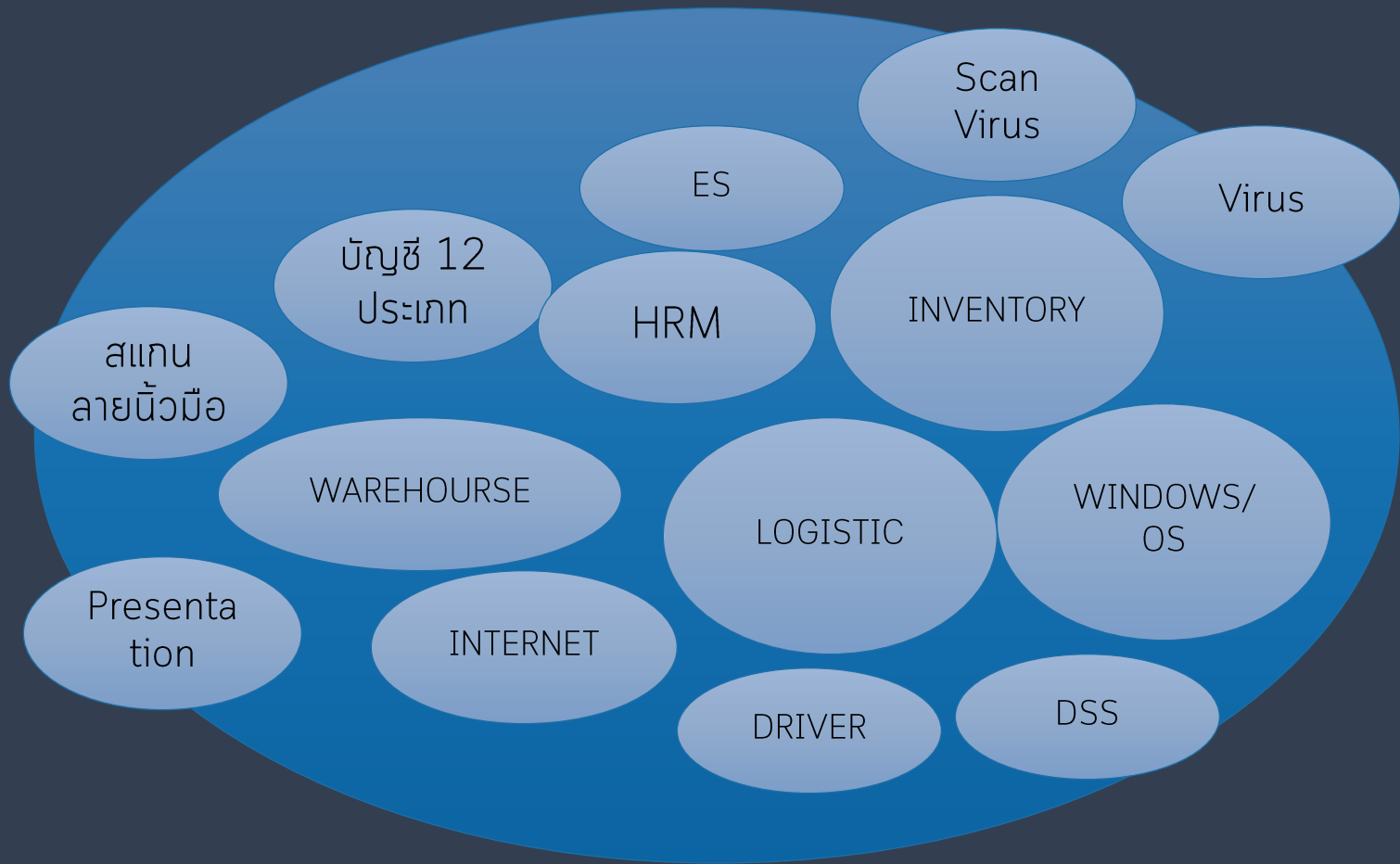
- เปิด เครื่อง
 - Boot - อุปกรณ์ต่อพ่วง Mouse, keyboard ..monitor
 - Ready ..window start up
 - Activate
 - Double Click ที่ icon ms-word load ข้อมูลของเวิร์ด อยู่ใน HD เข้าสู่แรม เพราะ HD ทำงานช้ากว่า CPU ทำงานระดับไมโครวินาที(ในวิชา OS) เข้าสู่ RAM ทำงานช้ากว่า CPU นาโน(10^{-9})s แต่ GHz. จึงต้อง แคช**รีจิสเตอร์** ...

ระบบปฏิบัติการ (OS: Operating System)

- Queue จัดการคิว

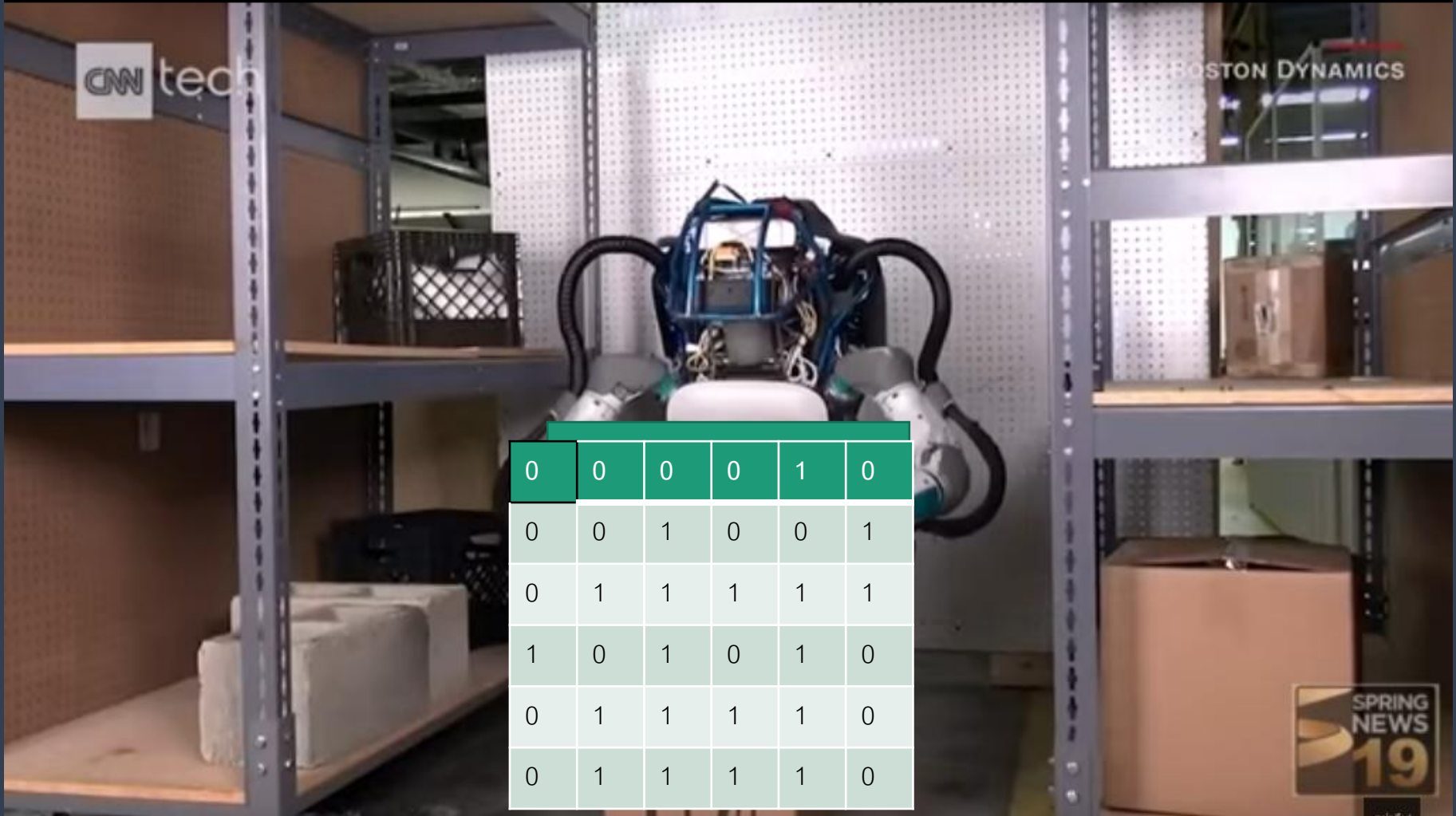


ซอฟต์แวร์ในอุดมคติ



จงอธิบาย ? 1 หน้ากระดาษ จากรูป

"แอตลาส" หุ่นยนต์ตัวแรกของโลก ที่ดีล้งกากลับหลังได้ | 30 พ.ย. 60



0	0	0	0	1	0
0	0	1	0	0	1
0	1	1	1	1	1
1	0	1	0	1	0
0	1	1	1	1	0
0	1	1	1	1	0

อย่าลืม!
SUBSCRIBE



0:09 / 1:29

เลื่อนเพื่อดูรายละเอียด
▼



โจทย์

- คอมพิวเตอร์เข้าใจภาษาอะไร ?
- ทำไมคอมพิวเตอร์เข้าใจแค่ภาษาเครื่อง
- หรือภาษาเครื่องคืออะไร
- ภาษา C คอมพิวเตอร์เข้าใจหรือไม่ ?

44 ตัว
26 ตัว
0-9
F1-F12

1000 ตัว
26 ตัว
0-9
F1-F12

10101110 = 8 Bit = 1 Byte = 1 ตัว → Ascii Code

0000 0000

...

1111 1111



256 แบบ

ระดับของภาษาคอมพิวเตอร์

1. ภาษาเครื่อง
2. ภาษาสัญลักษณ์ / Assembly
3. ภาษาชั้นสูง (High Level Language)
 1. Pascal ,Basic,GWBasic,Qbasic, C , C++ ฯลฯ
4. ภาษาชั้นสูงมาก (Very High Level Language)
5. ภาษาธรรมชาติ

01010010

Assembler

compiler

```
Mov AX,BX
ADD BX,2
SUB DX,2
```

```
Program CalCircle
Var R : integer;
Begin
  write("Enter R: ");
End.
```

SQL: Structure Query Language

compiler

interpreter

ภาษาเชิงวัตถุ
ภาษาแบบวิซวล

ส่ง Business Rule หรือ แผนธุรกิจ ของ บ. กลุ่ม

- siam2dev@hotmail.com
- ภายในอาทิตย์ ถัดไป
- ประกอบไปด้วย
 - ชื่อ บ. กลุ่ม
 - แผนธุรกิจ
 - สมาชิกในกลุ่ม
 - Present นำเสนอ WORD + PPT

ភាសាស័ព្ទលក្ខណ៍ / Assembly

- Depend on hardware → Assembly
- Independent → pascal , c

1. ภาษาเครื่อง

010101101111100111000111001111111100001010110111110
111000111001111111100001010110111110011100011100110
0111000111001111111100001010110111110011100011100
1100111000111001111111100001010110111110011100011
1001100111000111001111111100001010110111110011100
0111001100111000111001111111100001010110111110011
1000111001100111000111001111111100001010110111110
0111000111001100111000111001111111100001010110111
1100111000111001100111000111001111111100001010110
1111100111000111001100111000111001111111100001010
1101111100111000111001100111000111001111111100001
01011011111100111000111001100



A



BUG

DEBUG

SQL

- DBMS/RDBMS

- อยากได้ข้อมูลลูกค้าที่มีฐานเงินเดือนสูง 50000 ?

**“SELECT CustName , CustSalary FROM TbCustomer
WHERE CustSalary > 50000”**

Database Management System

Relational Database Management
System

ODBMS

Object Database Management
System

ถ้าเป็นโปรแกรมภาษาชั้นสูง

- เปิดไฟล์ฐานข้อมูล

Physical

- อ่านข้อมูลตั้งแต่แถวแรกไปจนเจอ
- เราต้องแสดงข้อมูลออกมา

SQL

- DML : Data Manipulation Language การจัดการกับข้อมูล
- **“SELECT * FROM TbMember”**
- **INSERT , UPDATE , DELETE**
- DDL Data Defination Lanuage
 - นิยามโครงสร้าง CREATE TABLE , CREATE DATABASE
- DCL Data Control Language
 - GRAINT

ภาษาธรรมชาติ NLP

- Natural Language Programming
 - Lisp
 - Prolog

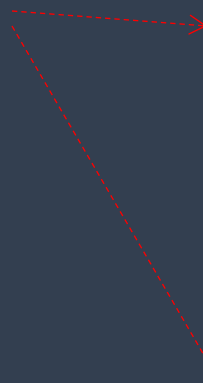
iRobot



Identity : อัตลักษณ์



Robot:**Dreaming**



AI : Artificial Intelligent
ปัญญาประดิษฐ์

Nueral Network
เครือข่ายประสาทเทียม

ตัวอย่างจากภาพยนตร์
Terminator : ศบเหล็ก
Skynet

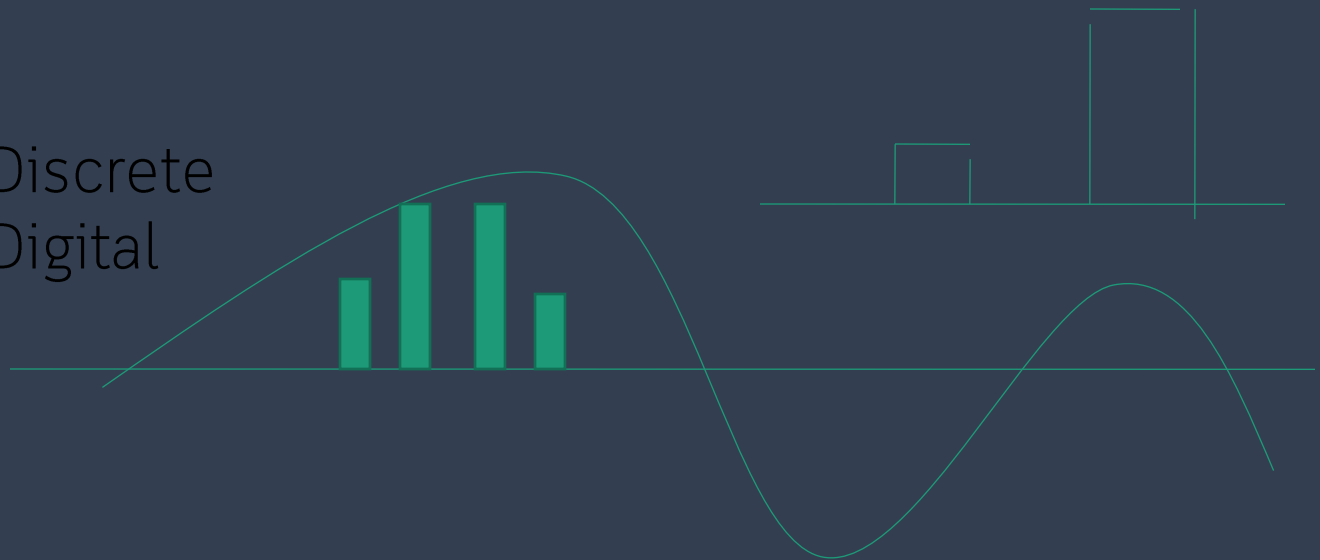
How Does Software Work?

ชุดคำสั่ง (Instructions) และ ข้อมูล (data) ร่วมกันทำงาน เพื่อสร้าง ลำดับของสถานะแบบไม่ต่อเนื่องภายในคอมพิวเตอร์

- อาจหลายสถานะที่สามารถเป็นไปได้ ที่เวลาใดๆ
- สถานะต่างๆ ของ Software ทำความเข้าใจได้ยาก

Analog

Discrete
Digital



Dim age as int16



ASCII code

- 0000 0000
- 1111 1111

จากวิกิพีเดีย สารานุกรมเสรี

แอสกี(ASCII) หรือ รหัสมาตรฐานของสหรัฐอเมริกาเพื่อการแลกเปลี่ยนสารสนเทศ ^[1] (อังกฤษ: ASCII: American Standard Code for Information Interchange) เป็น รหัสอักขระ ที่ประกอบด้วย อักขระละติน เลขอารบิก เครื่องหมายวรรคตอน และ สัญลักษณ์ ต่างๆ โดยแต่ละรหัสจะแทนด้วยตัวอักขระหนึ่งตัว เช่น รหัส 65 (เลขฐานสิบ) ใช้แทนอักษรเอ (A) พิมพ์ใหญ่ เป็นต้น

Systems and Software Systems

A System is

“An organized or connected group of Objects; a whole composed of parts in orderly arrangement according to some scheme or plan”

A software-intensive System (Software System) is

system ที่ประกอบด้วยระบบ software ระบบย่อย ที่มีบทบาท หรือมีส่วนในการควบคุมส่วนสำคัญของพฤติกรรมทั้งหมดของระบบ

OS

Examples of Software Systems

ระบบควบคุมการบิน, ระบบทางรถไฟ, ระบบซื้อขายหลักทรัพย์, ระบบบัญชี, ระบบงานธนาคาร, ระบบการเรียนการสอน ฯลฯ

"Why" Do We Build Software Systems?

- เพื่อแก้ปัญหาตามความต้องการของผู้ใช้(User)
 - software ต้องสามารถแก้ปัญหของผู้ใช้ได้
 - software ต้องสามารถทำงานที่ผู้ใช้ต้องการได้อย่างถูกต้องและมีประสิทธิภาพ

แท้จริงแล้วซอฟต์แวร์ คือสัญญาณไฟฟ้า



MIS



MIS

“How” Do We Build Software Systems?

- โดย “decomposition” และ “synthesis”
 - จำแนกย่อย และสร้างแบบจำลองของปัญหาในรูปแบบที่สามารถติดต่อสื่อสารกับผู้อื่นได้ (กระบวนการ Specification/Analysis)
 - ข้อมูล/สารสนเทศเกี่ยวกับขอบเขตของปัญหา (problem domain) จะต้องเป็นที่เข้าใจตรงกัน
 - ทำการสังเคราะห์ (Synthesize) แนวทางการแก้ปัญหา (solution) โดยมุ่งเน้นที่ actions หรือ data (กระบวนการ Design/Implementation)

Divide-
conquer

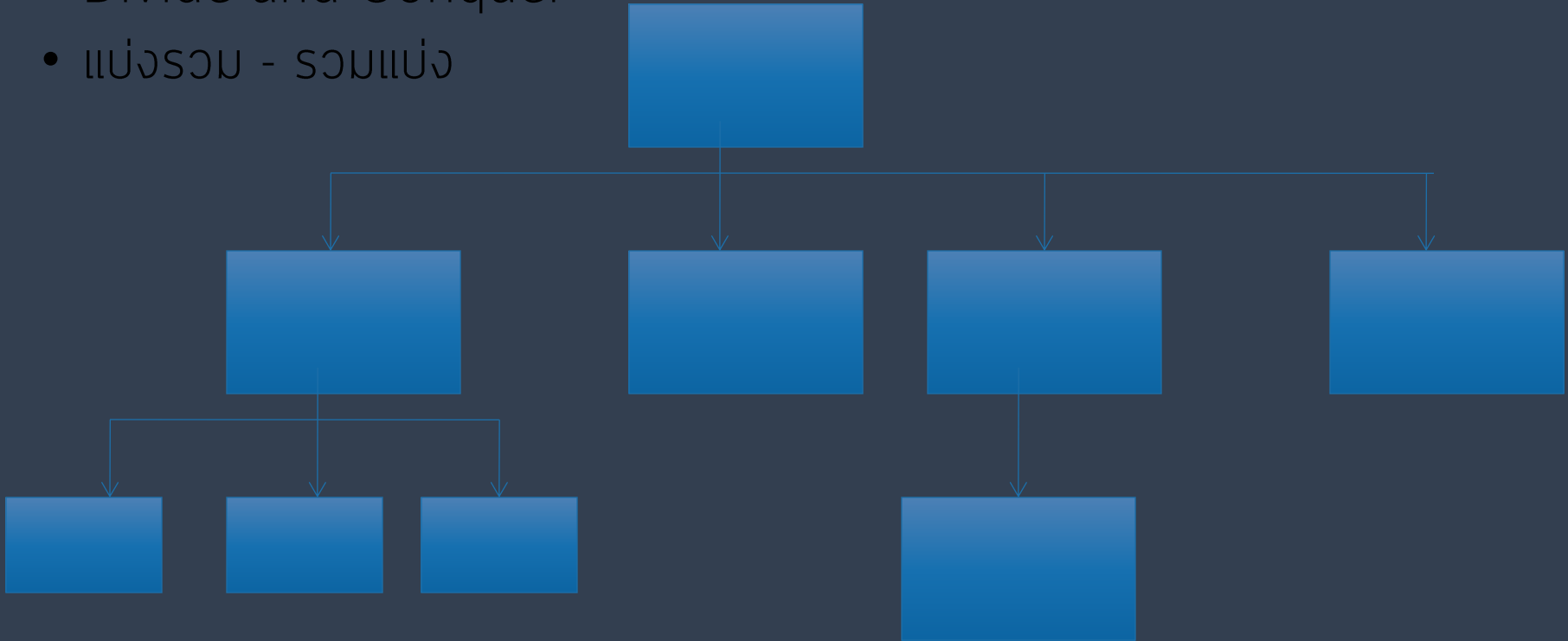
Top-Down



Structure Chart

- Divide and Conquer
- ແບ່ງຈັບ - ຈັບແບ່ງ

Integrate System



ตัวอย่างการออกแบบจากบนลงล่าง Top-down design

Microsoft Office

```
graph TD; MO[Microsoft Office] --> MW[Microsoft Word]; MO --> ME[Microsoft Excel]; MO --> MA[Microsoft Access]; MO --> MP[Microsoft Power Point]; ME --> WB[Workbooks]; WB --> WS1[Worksheet 1]; WB --> WS2[Worksheet 2]; WB --> WS3[Worksheet 3]; WB --> WSN[Worksheet N];
```

Microsoft Word

Microsoft Excel

Microsoft Access

Microsoft Power
Point

Workbooks

Worksheet
1

Worksheet
2

Worksheet
3

Worksheet
N

แบบฝึกหัด

- จงยกตัวอย่างและเขียนภาพประกอบซอฟต์แวร์ที่ใช้หลักการ Divide-conquer มาและอธิบายสังเขป
- จงยกตัวอย่างและเขียนภาพประกอบซอฟต์แวร์ที่ใช้หลักการ Divide-conquer มาและอธิบายสังเขป
 - โปรแกรมจัดการคลังสินค้า
 - โปรแกรมบริหารศูนย์บริการรถยนต์

Software Development Process

- **Requirement Specification** : define problem domain
- **Analysis** : what problem to be solved?
 - เพื่อบอกว่ามีปัญหาอะไรบ้าง ปัญหา 1 ปัญหา 2 ปัญหา 3 ปัญหา ... ที่ต้องแก้
- **Design** : how to solve the problem?
 - แต่ละปัญหาแก้อย่างไร
- **Implementation** : how to implement the solution?
 - ทำตามที design ไว้
- **Testing** : how to ensure that the solution can solve the problem?
- **Maintenance** : how to adjust the solution to accomodate change?
- **Retirement** : when does the system to be retired?

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

What is Software ?

Requirement

- Requirement >> โปรแกรมสามารถคำนวณต่าง ๆ ดังนี้ได้
 - หาค่าพื้นที่ต่างๆ
 - คำนวณ ค่าเฉลี่ยได้
 - คำนวณ SQRT
 - คำนวณ POWER
- Analysis
 - 1. หาค่าพื้นที่ต่างๆ
 - 2. คำนวณภาษีได้

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

What is Software ?

- Flowchart
- Algorithm
 - Begin/Start
 - ...
 - ...
 - ...
 - ...
 - ...
 - End/Stop

ค่าพื้นที่ต่างๆ

START

STOP

FLOWCHART

- SYSTEM FLOWCHART
 - ทั้งระบบ
- PROGRAM FLOWCHART
 - ส่วนโปรแกรม

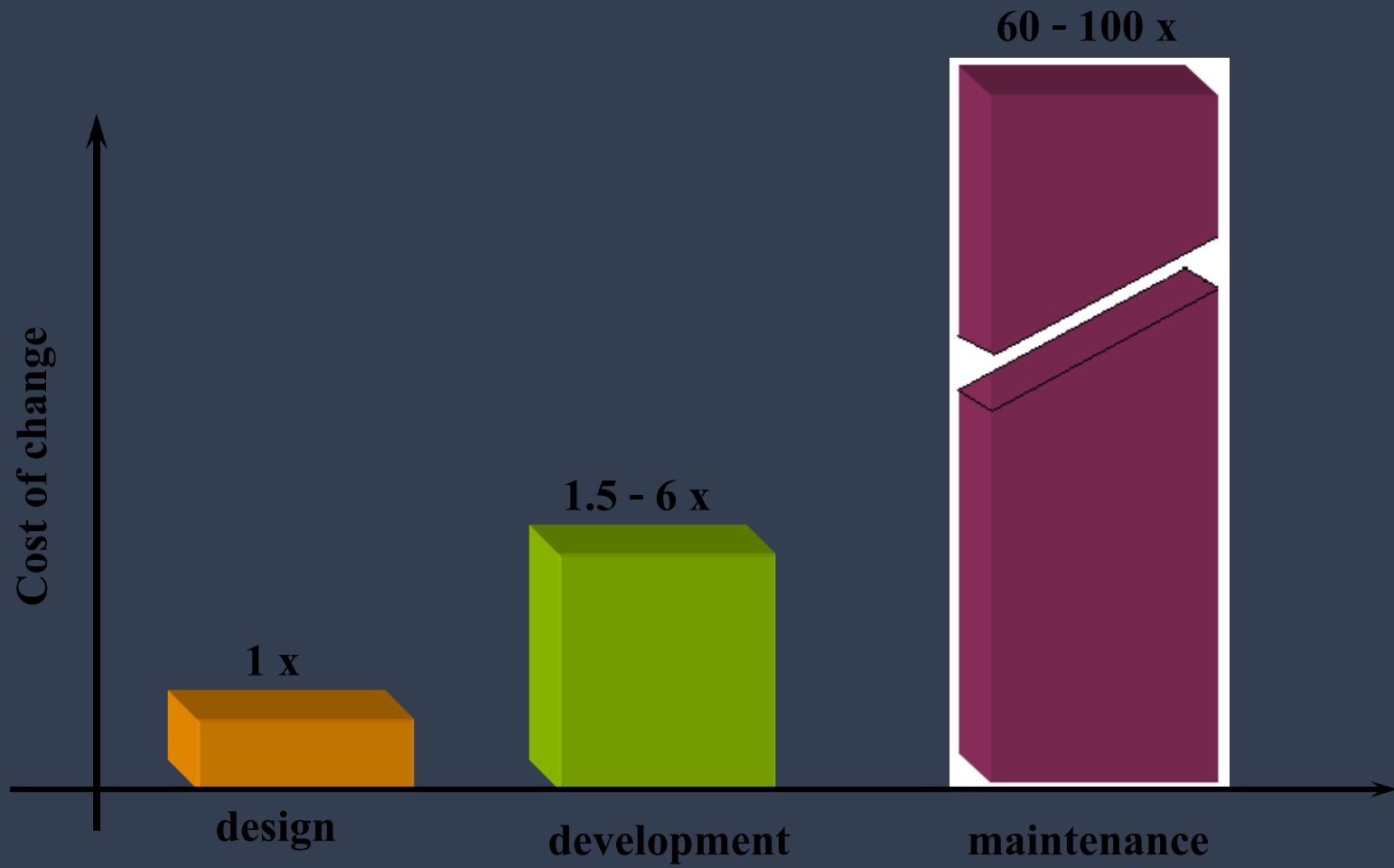
*ผังงานระบบ และ ผังงานโปรแกรม เหมือนหรือ
ต่างกันอย่างไร ?
ให้ยกตัวอย่าง*

Software Development Life Cycle

- SDLC Model

- ประกอบด้วยจำนวนชุดของขั้นตอน ในกระบวนการพัฒนา Software (Software Development Process)
- อาจเกิดปัญหาได้ในทุกๆ ขั้นตอนการพัฒนา
- มีหลาย Model ที่แตกต่างกันในวิธีดำเนินการของขั้นตอน เช่น **Waterfall Model, Incremental Model, Spiral Model** เป็นต้น
- แต่ละ Software ใช้ Model ที่แตกต่างกันในการพัฒนา
- แต่ละ Model มีจุดเด่น จุดด้อย ในตัวเอง

The Cost of Change



ความเสี่ยง หรือ ค่าใช้จ่ายในการพัฒนาระบบเมื่อเวลาผ่านไปเรื่อยๆ จะสูงแบบทวีคูณ

“What Techniques” Can Be Used?

- ขึ้นอยู่กับสิ่งที่ software engineer มุ่งเน้นระหว่างการสร้างเครื่อง (Synthesis) แนวทางการแก้ปัญหา
 - มุ่งเน้น **Process (Action)** ...งาน หรือ ฟังก์ชันของระบบ
 - มุ่งเน้น **Data** ข้อมูล..
 - มุ่งเน้น **Object** ...ออบเจกต์....

Traditional Software Development

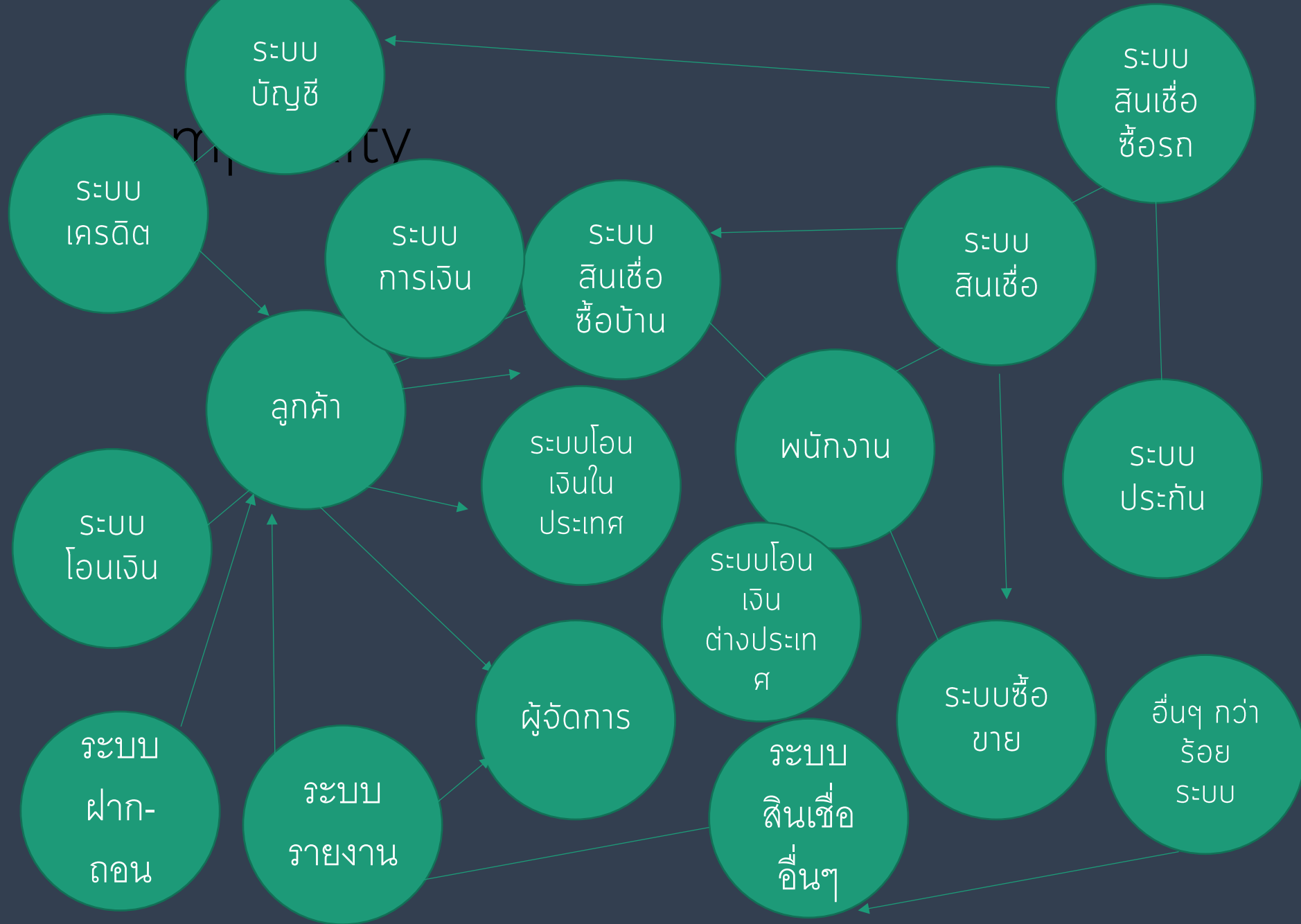
- ใช้ Structured Analysis & Design Techniques
 - พิจารณาเฉพาะ Data (Data-Oriented)
 - ตัวอย่างเช่น Data Flow Model, ER (Entity-Relational) Model
 - พิจารณาเฉพาะ Action (Action-Oriented) ที่เกี่ยวเนื่อง/กระทำกับ Data
 - ตัวอย่างเช่น Finite State Machine (FSM)
- ใช้ Structured Implementation Techniques
 - ใช้ Structured/Procedural Programming Languages
 - ตัวอย่างเช่น BASIC, Pascal, C

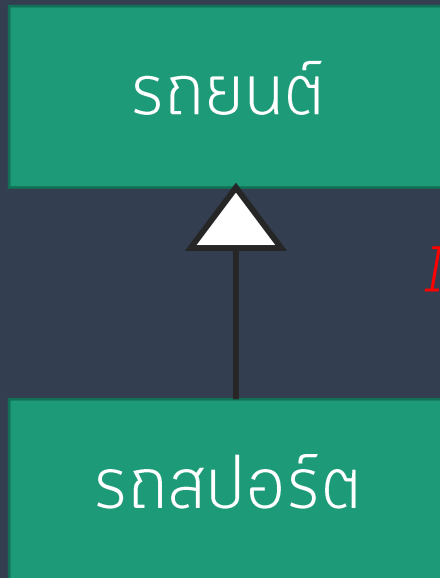
OO vb vc# java Delphi

Hybrid บางภาษายังทำงานแบบดั้งเดิม หรือแบบโครงสร้าง

Problems with Traditional Techniques

- Do Traditional techniques work?
 - Yes, they do (at a cost !!!)
- Problems
 - ในปัจจุบัน Commercial Software จำนวนมาก ยังคงใช้ Traditional Techniques
 - Software มีขนาดใหญ่ และซับซ้อนมากขึ้น เพื่อตอบสนองความต้องการของผู้ใช้และราคาที่ถูกลงของ Hardware
 - การค้นหาข้อบกพร่องของ Software Design และ Implementation ใช้เวลานาน และเสียค่าใช้จ่ายสูง
 - การบำรุงรักษา Software ทำได้ไม่สะดวก และมีค่าใช้จ่ายเข้ามาเกี่ยวข้อง





Inheritance

4 ประตู

มีล้อ

มีเครื่องยนต์

4 ประตู

มีล้อ

มีเครื่องยนต์ เทอร์โบ

เปิดประทุน

- **Reusable** คือ การนำกลับมาใช้ใหม่ ตามหลักการของ OO โดยทำงานใหม่ที่เพิ่มเข้ามาจะไม่ส่งผลกระทบต่องานเดิม

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

Software มีการเปลี่ยนแปลง (Change) ตลอดเวลา

- Software มีการเปลี่ยนแปลง (Change) ตลอดเวลา
 - ระบบ Software เดิม (Legacy Software) ไม่สามารถปฏิบัติงานร่วมกันกับเทคโนโลยีแบบใหม่ได้ (Interoperability)
 - ยากแก่การขยายขอบ (Extendability)
 - ไม่สามารถนำ Software ไปใช้ซ้ำ (Reusability)
- มักพัฒนา Software จากเริ่มต้น ไม่ได้พัฒนาจากของเดิมที่มีอยู่

ปัญหาของการพัฒนาระบบ Software เดิม
แบบเดิม
จึงนำไปสู่การพัฒนาโดย
หลักการเชิงวัตถุ

Software Crisis

- การพัฒนา **Hardware** เร็วมาก
 - ขนาด (**Size**) ของซอฟต์แวร์ ใหญ่ขึ้น และเพิ่มความซับซ้อน (**Complexibility**) มากขึ้น
- แต่การพัฒนา **Software** ช้า และมี **bug**
 - ค่าใช้จ่าย (**Cost**) ในการพัฒนา และบำรุงรักษาสูง (**Maintainability**)
 - ใช้เวลาในการพัฒนา (**Time to Market**) นาน หรือ ได้ซอฟต์แวร์คุณภาพด้อยที่ไม่ตรงกับความต้องการของผู้ใช้
- ช่องว่างระหว่าง **Users** และ **Programmers**

What are Object-Oriented Techniques?

- มุมมองใหม่ของ Software Decomposition
 - พิจารณาข้อมูล (attribute/data) และ หน้าที/พฤติกรรม (Action/behavior) รวมกันเป็นวัตถุ (Object)
- เสนอแนวทางใหม่ในการ Synthesis
 - พิจารณาแนวทางการสังเคราะห์ วิธีการแก้ปัญหา โดยมุ่งเน้นที่ Data และ Actions โดยรวมเป็นหน่วยเดียวกันคือ Object
 - แยกส่วนของ Data และ Actions ที่ไม่เกี่ยวข้องไว้ ในรูปของ การเรียกใช้ Module
- พิจารณาขอบเขตของปัญหา (Problem Domain) และสร้างแบบจำลองการแก้ปัญหา (Modeling Objects) ตามสภาพความเป็นจริง (Real World)

ปัญหาของการพัฒนาระบบแบบเดิม
จึงนำไปสู่การพัฒนาโดยหลักการเชิงวัตถุ

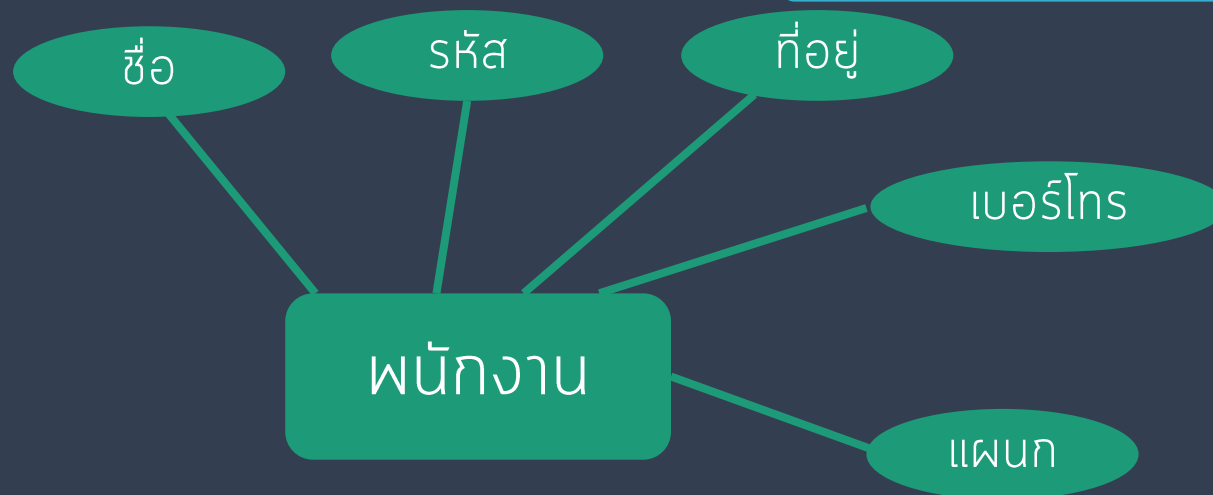
มุมมองใหม่ของ Software Decomposition

พิจารณาข้อมูล (attribute/data) และ หน้าที/พฤติกรรม (Action/behavior) รวมกัน
เป็นวัตถุ (Object)

- มองพนักงาน แบบดั้งเดิม เป็นเพียง data หรือ entity ธรรมดา มีแค่
ข้อมูล เช่น มีชื่อพนักงาน รหัส เบอร์โทร อีเมล

เราเลือก อะไรเป็น PK ? เหตุผล

ทำไมจึงไม่เก็บแผนกในพนักงาน ?



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

ตารางพนักงาน

ตารางพนักงาน

ID	Name	Gender	Tel	Address	Department	Salary
001	ชโลธร	Female	0888777711	BKK	แม่บ้าน	2500
002	ชาลิสา	Male	0811	กกน	สปท.	180

Anomaly

Insert Anomaly

Delete Anomaly

Update Anomaly

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

ตารางพนักงาน

ID	Name	Gender	Tel	Address	Dept_ID	Salary
001	ชโลธร	FeMale	0888 7777 11	BKK	D001	2500
002	ชาลิส่า	Male	0811	กทม	D002	180

Dept_ID	Dept_Name	Detail
D001	แผนกบ้าน	
D002	รปภ.	

Anomaly
Insert Anomaly
Delete Anomaly
Update Anomaly

Primary Key หมายถึง

เราเลือก อะไรเป็น PK ? เหตุผล

ให้ยกตัวอย่าง อะไรที่สามารถเป็น ค่า Null ได้

1. สิ่งที่เป็น unique ไม่ซ้ำ
2. จะแทนข้อมูลทั้งกับเพิล Tuple ได้ หรือ แถว หรือ เรคคอร์ด
3. จะต้องไม่ว่าง null ไม่ได้
4. จะสื่อความหมาย และสั้นๆ ได้ใจความ
5. จะต้องไม่เป็น Multi Value Attribute

ชื่อ เบอร์โทร

ให้ยกตัวอย่าง อะไรที่สามารถเป็น ค่า Multi Value Attributes ได้

เพราะอะไร จึงไม่เลือกที่อยู่เป็น PK

Multi Value Attribute

นาย ก อยู่ กทม
นาย ข อยู่ กทม
นาย ง อยู่ กทม

นาย ก อยู่ กทม
นาย ก อยู่ เชียงใหม่
นาย ก อยู่ สงขลา

นาย ก เบอร์ 081-xxxxxxx
นาย ก อยู่ 02-
นาย ก อยู่ 089-

E-mail

ID	Name	Gender	Tel	Address	Dept_ID	Salary	E-mail
001	ชโลธร	FeMale	088 877 771 1	BKK	D001	2500	salothon1@gmail.com salothorn2@Hotmail.com
002	ชาลีสา	Male	081 1	กทม	D002	180	

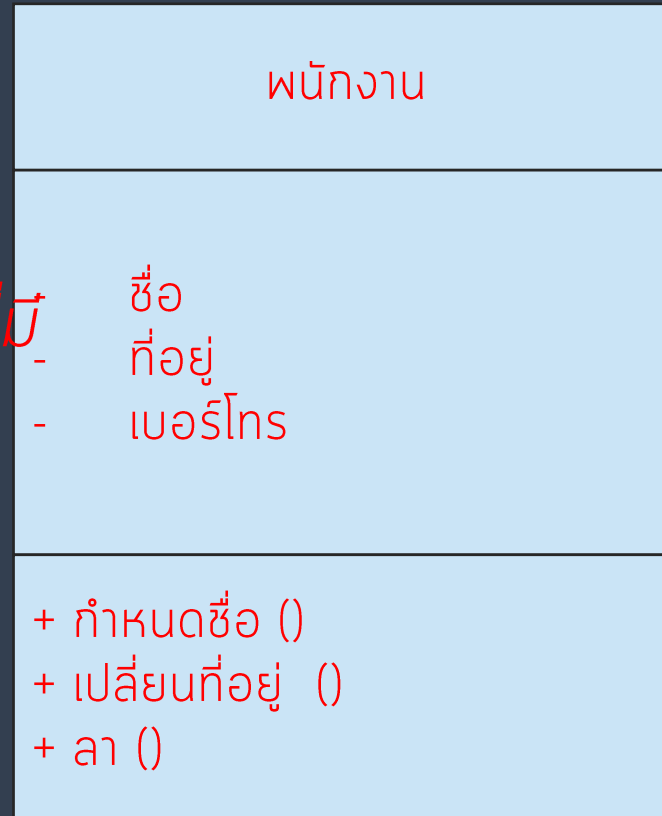
ใน หลักการเชิงวัตถุ จะมองพนักงานเป็น object

- มองพนักงานใหม่ในเชิง OO

*** มอง พนักงาน เป็นวัตถุ ที่มี
Attribute และฟังก์ชัน

Attribute

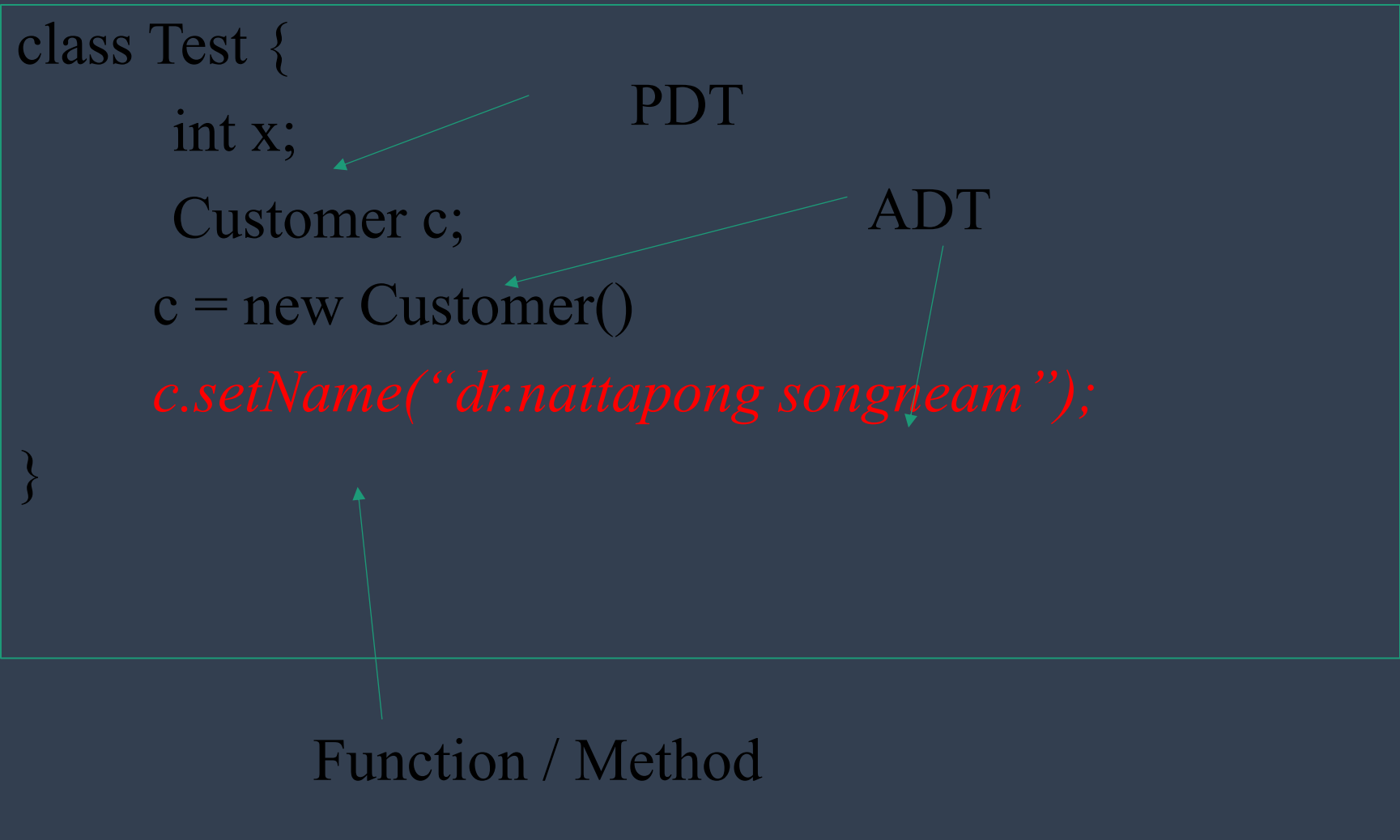
Function



ต.ย. โปรแกรมในภาษาจาวา

```
class Test {  
    int x;  
    Customer c;  
    c = new Customer()  
    c.setName("dr.nattapong songneam");  
}
```

PDT

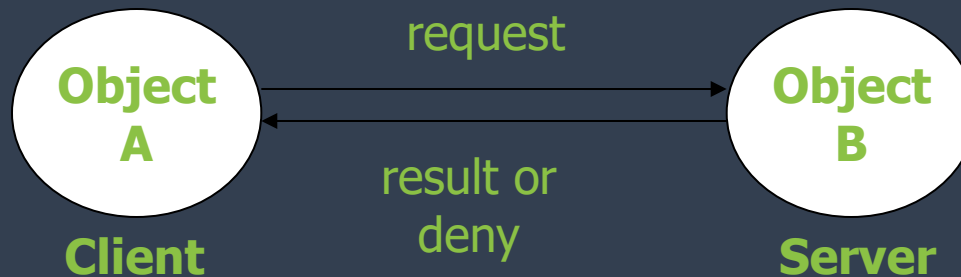


ADT

Function / Method

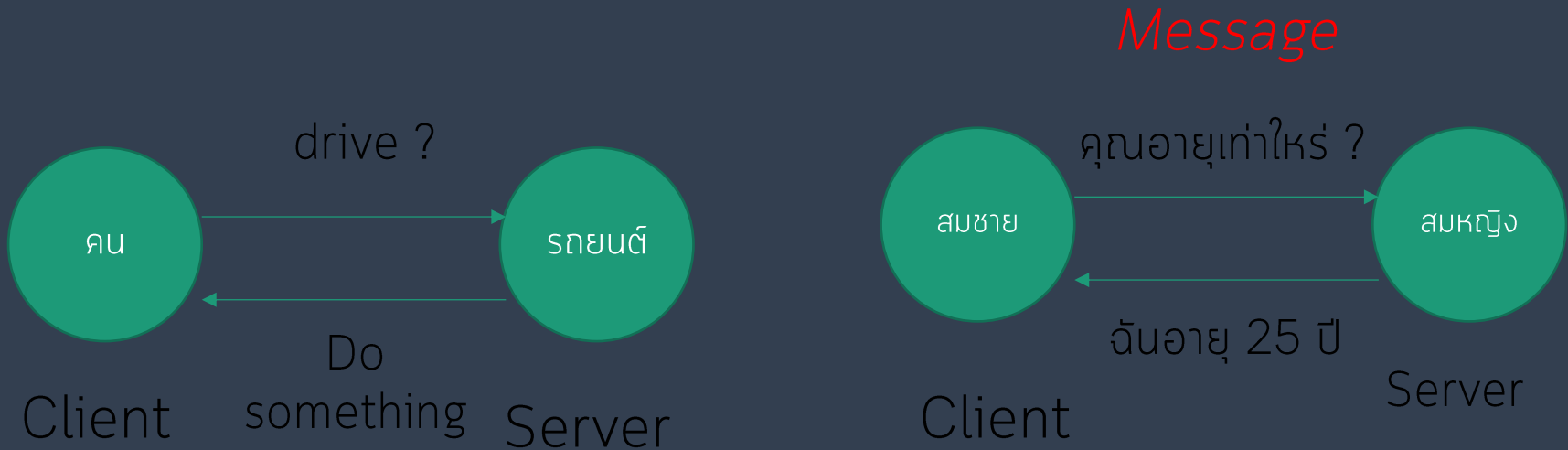
What are Object-Oriented Techniques?

- พิจารณา software ในรูปแบบของ **clients** และ **servers**



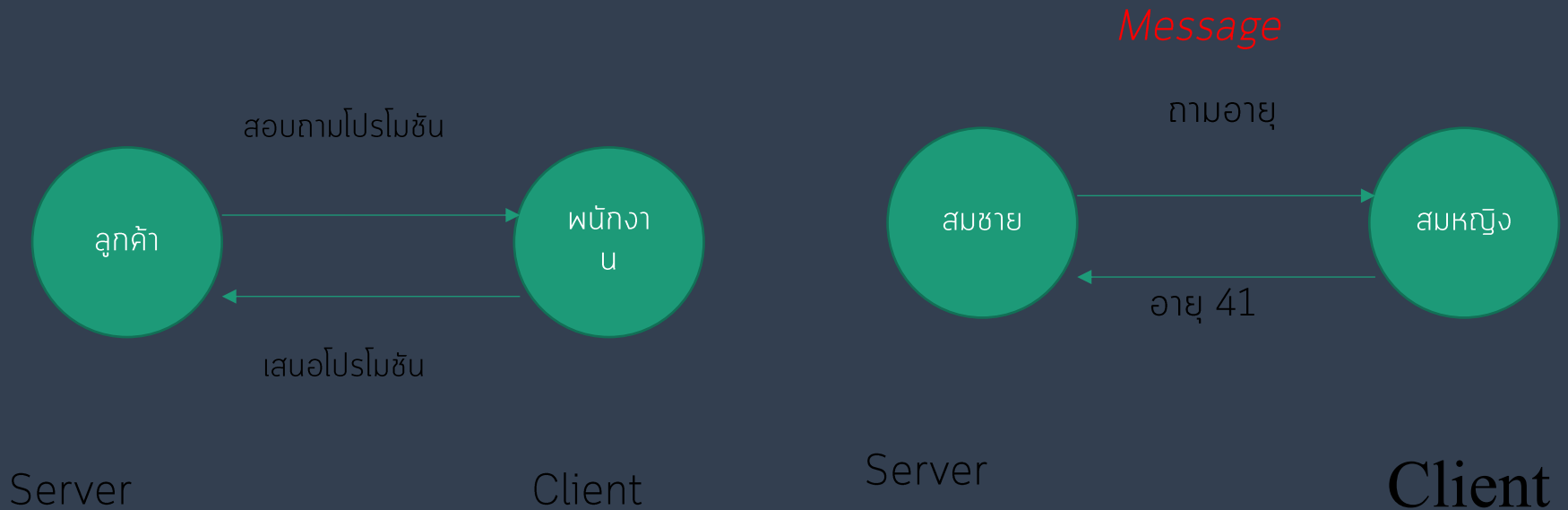
- servers** ซ่อน Implementation ของ **Actions** ไว้ภายใต้ **Interface**
 - clients** เรียกใช้ได้เฉพาะ **Interface** ที่ **server** เปิดให้บริการเท่านั้น
- *** Server มีความสามารถบางอย่างที่ให้ client เรียกใช้ได้*

ในซอฟต์แวร์เชิงวัตถุ จะมี object หลายตัวที่ทำงานร่วมกัน



*ตัวที่กระตุ้น หรือเริ่มทำงาน มองเป็น server
ตัวที่ตอบกลับมา หรือ ตอบสนอง เป็น client*

ในซอฟต์แวร์ เชิงวัตถุ จะมี object หลายตัวที่ทำงานร่วมกัน

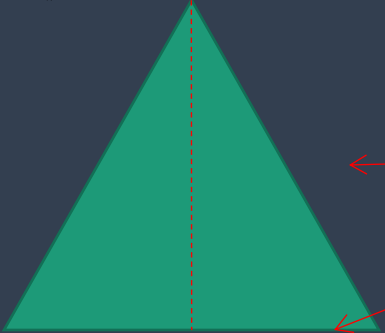


ตัวที่กระตุ้น หรือเริ่มทำงาน มองเป็น server
ตัวที่ตอบกลับมา หรือ ตอบสนอง เป็น client

class

```
T t1 = new T();
```

Object



Base
Hieght

```
setBased()  
calArea()  
t1.calArea()
```

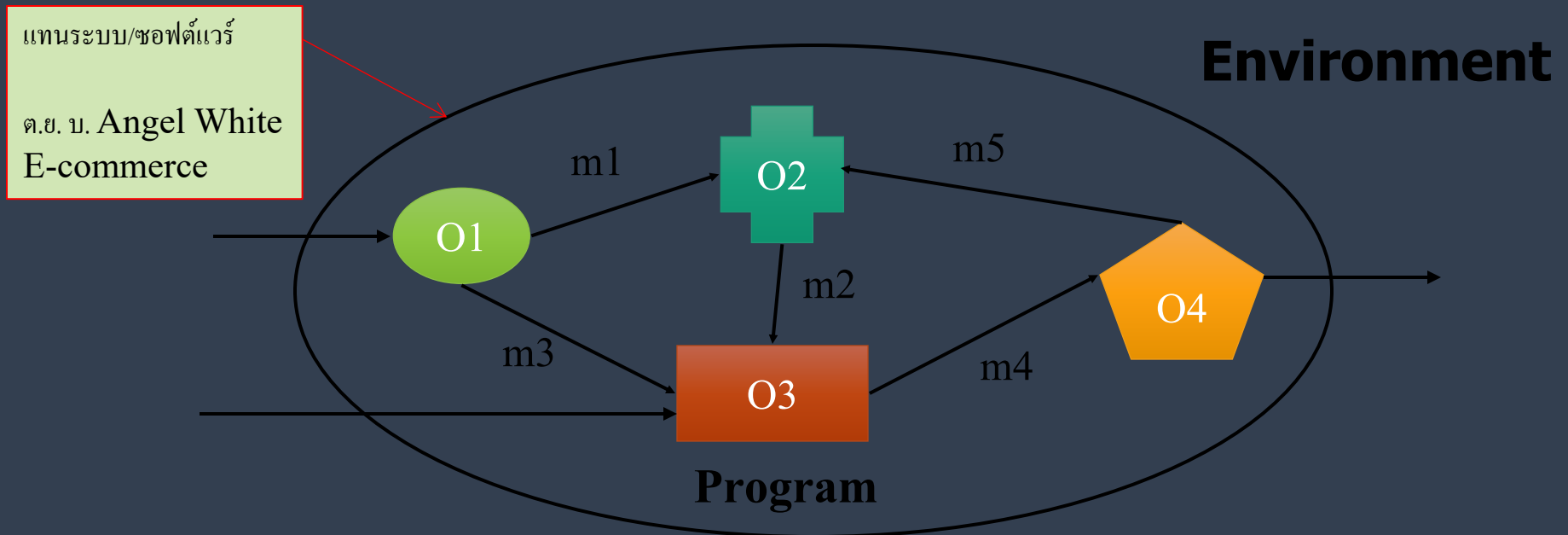
Attributes / **Data Member**
Function/Behavior / **Method Member**

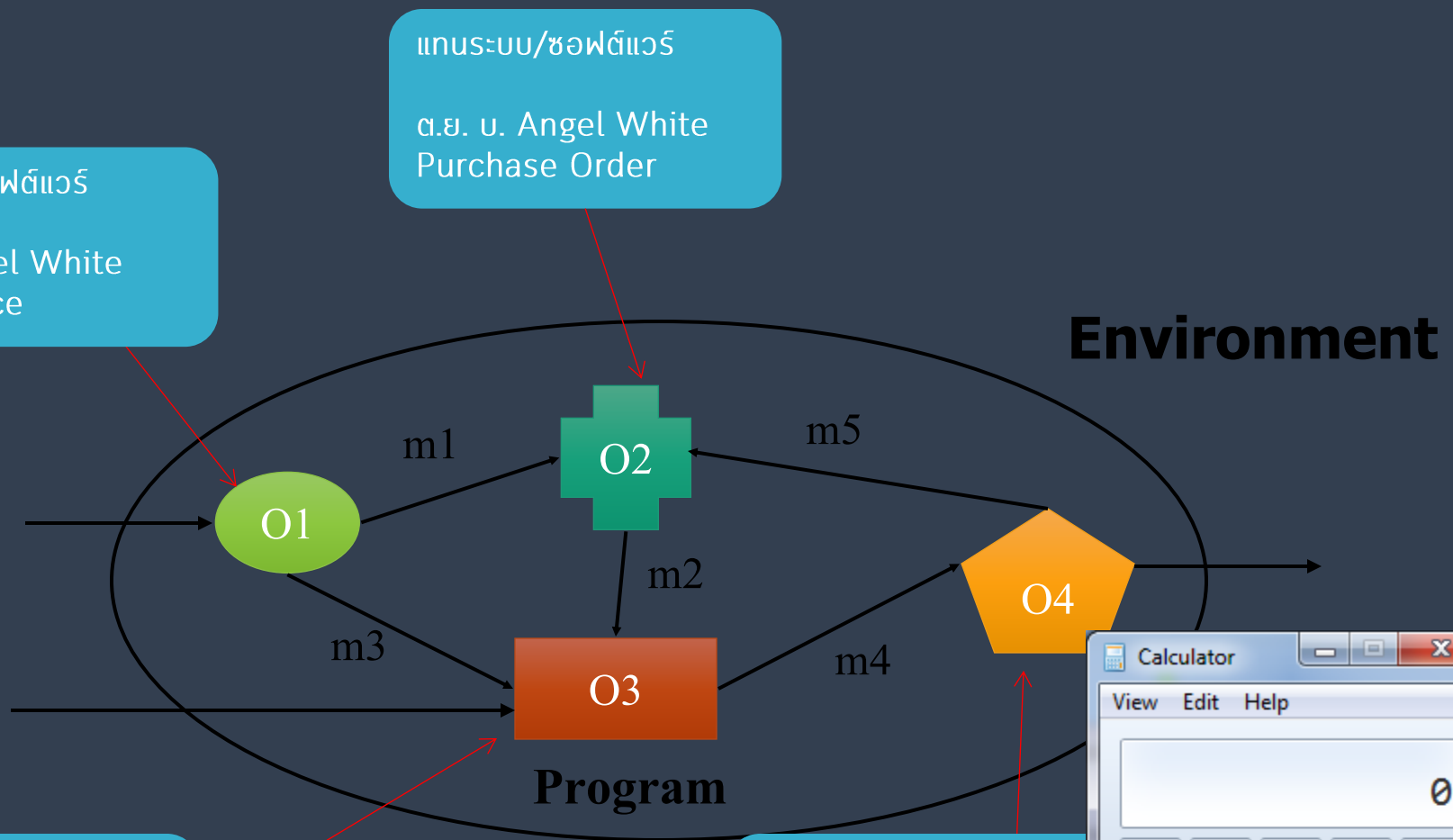
ใน OO

ในภาษาจาวา

Object-Oriented Software

- ซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software)
คือ กลุ่ม หรือชุด(Collection) ของวัตถุที่มีหน้าที่รับผิดชอบต่อการจัดการข้อมูลของตนเอง และติดต่อสื่อสารกับวัตถุอื่นๆ โดยส่งข้อความ (Message) ให้แก่กันและกัน





แอมระบบ/ซอฟต์แวร์

ด.ม. บ. Angel White
Purchase Order

แอมระบบ/ซอฟต์แวร์

ด.ม. บ. Angel White
E-commerce

แอมระบบ/ซอฟต์แวร์

ด.ม. บ. Angel White
HRM

แอมระบบ/ซอฟต์แวร์

ด.ม. บ. Angel White
Inventory

1

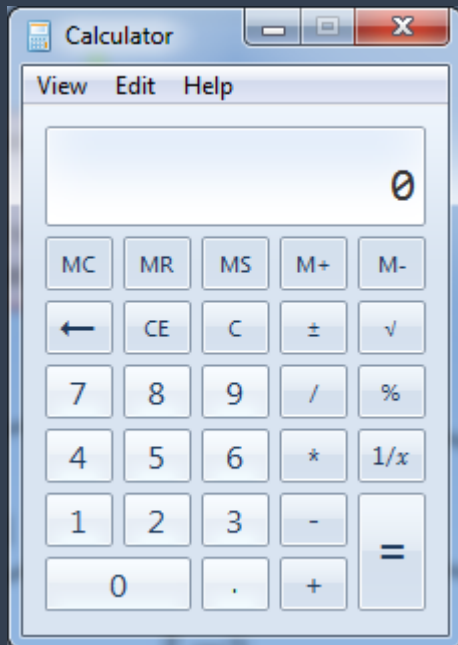
บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

Object-Oriented Programming



class

object

- Button ...b1b9 b0
- Menu file object
- Textbox t1

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

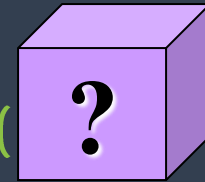
บทที่ 1

OO : *What is an object?*

- วัตถุ (object)

เป็น กลุ่มหรือชุด (collection) ของสิ่งต่อไปนี้

- แอททริบิวต์ (Attribute) ของวัตถุจะบอกถึงข้อมูล (คุณสมบัติเฉพาะหรือ สถานะของวัตถุ
- เมธอด (Method) จะบอกถึงหน้าที่ (functionality) หรือ พฤติกรรม (behavior) ต่างๆของวัตถุนั้นๆ



- โดยมี แอสโซซิเอชัน (Association) อันได้แก่ สิ่งที่บอกถึงความสัมพันธ์ (Relationship) ของวัตถุนั้น กับวัตถุอื่นๆ รวมทั้งปฏิสัมพันธ์ (Interactions) ระหว่างวัตถุกับวัตถุอื่นๆ



ปากกา

Implement in java

- Attributes

- สี
- ความยาว
- น้ำหนัก
- ฝาปากกา
- ราคา

- Actions/Functions

- เขียนได้

```
class PEN {  
    private String color;  
    private int width;  
    private double weight;  
    private double price;  
    private Object cap; //ฝาปากกา  
    public void Writable() {  
        ...  
    }  
    public void Openable() {  
        ....  
    }  
}
```

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

OO : *What is an object?*

ปากกา

attribute

- สี
- ไล่
- ฝา
- หัวปากกา
- ราคา
- ยี่ห้อ

function

- เขียนได้
- เปิดฝาปากกาได้
- หมุนได้



association

- ใส่ปากกา
- ฝาปากกา
- หมึก

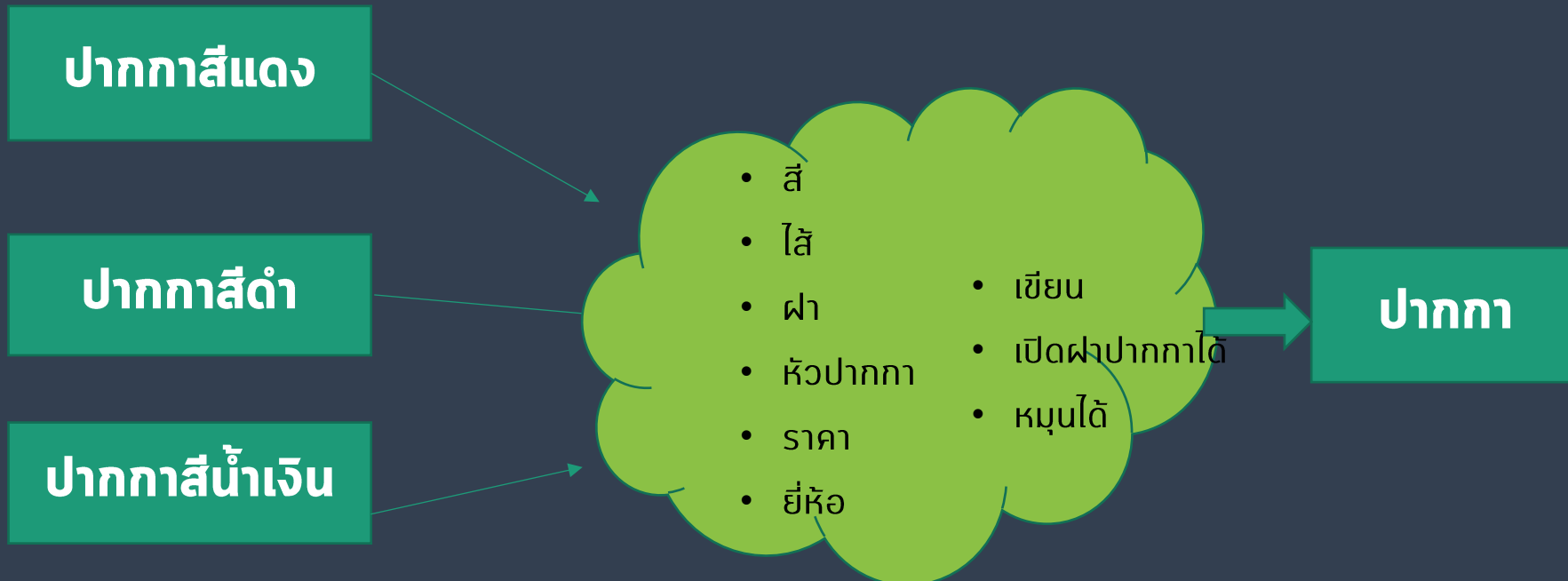
1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

OO : *What is an object?*



การให้ความคิดรวบยอดเกี่ยวกับปากกา

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

OO : *What is an object?*



การให้ความคิดรวบยอดเกี่ยวกับปากกา

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

OO : *What is an object?*

Functions/Behaviors/Methods/Actions

- สิ่งนั้นทำอะไรได้บ้าง คือความสามารถของสิ่งนั้น **Methods**
- สิ่งนั้นมีพฤติกรรม เป็นอย่างไร **Behaviors**
- สิ่งนั้นมีหน้าที่อะไร **Functions**

ในภาษาเชิงวัตถุ OOP

ในหลักการเชิงวัตถุ OO

1

บทที่ 1

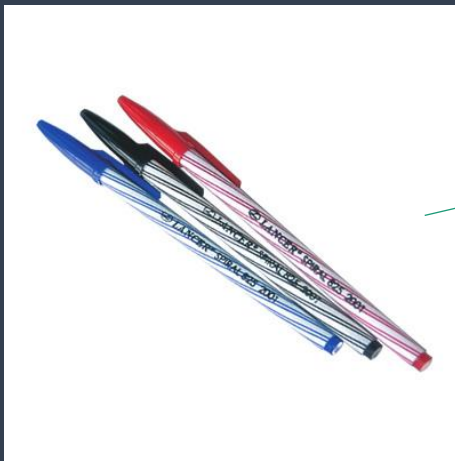
หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

OO : *What is an object?*

คุณลักษณะ / attributes

- สิ่งที่เรามองวัตถุ ว่า วัตถุนั้นมีอะไรบ้างเป็นส่วนประกอบ และมีหน้าตาเป็นอย่างไร



สี

หมึก

ด้าม

ฝาปากกา

ไส้

ราคา

ยี่ห้อ

ปากกา

Mobile



• Attribute

- รูปทรง
- สี
- ราคา
- น้ำหนัก

• Function/Action/Behavior

- โทรออก
- รับสาย
- เล่นเกม
- ดูราคา
- ดูรุ่น

```
class Mobile {
    private String Shapes;
    private Color color;
    private double price;
    private double weight;

    public void call() {

    }

    public void receive() {
        ...
    }

    public void setPrice(double p) {
        price = p;
    }

    public double getPrice() {
        return price;
    }
}
```

โทรออก ()
รับสาย ()
ส่งข้อความ ()



mobile



โทรออก ()
รับสาย ()
ส่งข้อความ ()
เล่นเกม ()
เล่นเฟส ()



smartphone



surface

โทรออก ()
รับสาย ()
ส่งข้อความ ()
เล่นเกม ()
เล่นเฟส ()
เขียนโปรแกรมได้ ()

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

OO : *What is an object?*

- ทวี
- ดินสอ
- รถยนต์
- รถเมล์
- รถไฟฟ้า
- รถไฟ
- คน
- แมว
- สุนัข

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

00 : *Identify Problem Domain*

Identify Problem Domain

- ระบุ ขอบเขตของปัญหา (Problem Domain) เพื่อสร้างแบบจำลองวัตถุ (Modeling Objects)
 - เป็นการกำหนดขอบเขตของสิ่งที่เราจะพิจารณาให้เป็นวัตถุ (Object)
 - เราไม่สามารถสนใจทุกๆ วัตถุ (Object) ในโลกของความเป็นจริง ในเวลาเดียวกันได้
 - เช่นเดียวกับที่เราไม่สามารถสนใจทุกความสัมพันธ์ และ กิจกรรม ทุกความสัมพันธ์ และทุกกิจกรรมที่เกิดขึ้นในโลกของความเป็นจริงได้เช่นกัน

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

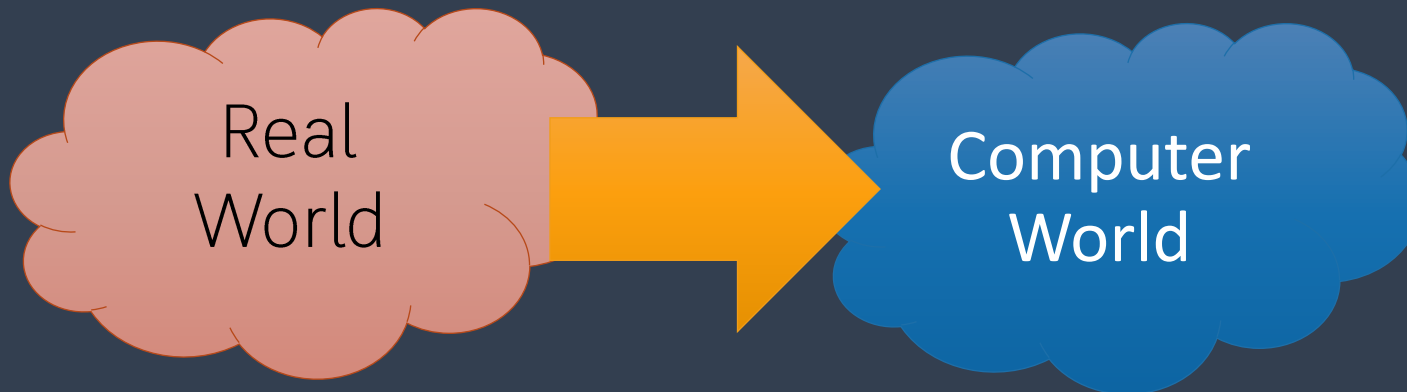
OO : *Real World / Computer World*

• รถในโลกความเป็นจริง เราจะนึกถึง

- ล้อ
- กระจก
- กระโปรงหน้า
- พวงมาลัย

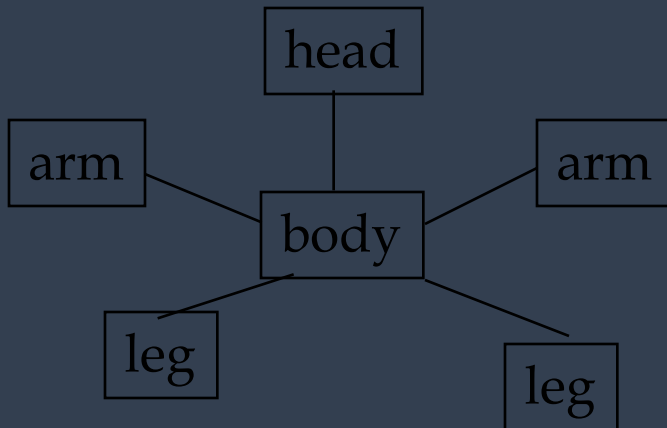
• รถในโลกของคอมพิวเตอร์ เราจะสนใจแต่

- ยี่ห้อ
- รุ่น
- ราคา
- สี



Objects = Attributes, Methods, Associations

Tim



Attributes

Age
Gender
Name
....

Associations

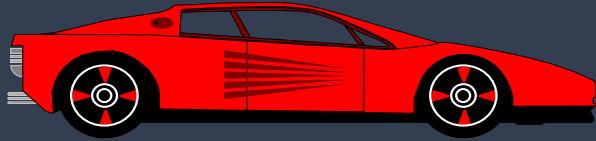
myLeftArm
myLeftLeg
myRightArm
....

Methods/Operations

Jump..
Talk...
Dance
Ask Friend
Jump
Walk
Jump
Shake myHead
Shake myBody

*A working OO system:
a conversation between networks
of objects*

Objects = Attributes, Methods, Associations



รถยนต์

Attributes

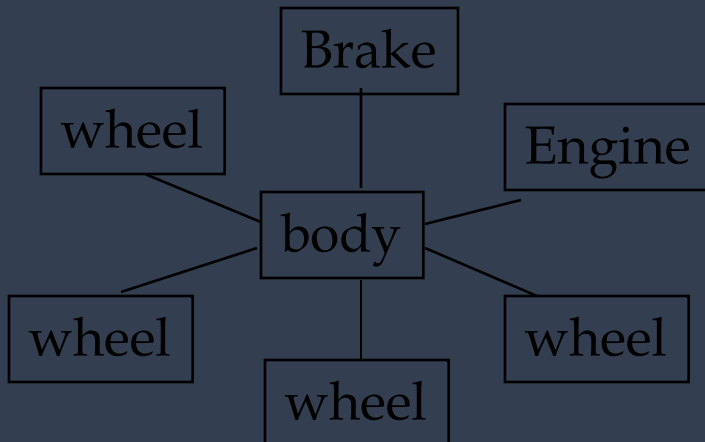
Model
Company
Color
....

Associations

Brake
LeftFrontWheel
Gear
....

Methods/Operations

Start..
Go..
Stop..
Go Forward
Go Bakcward
Turn Left
Turn Right
Apply Brake
Change Gear
....

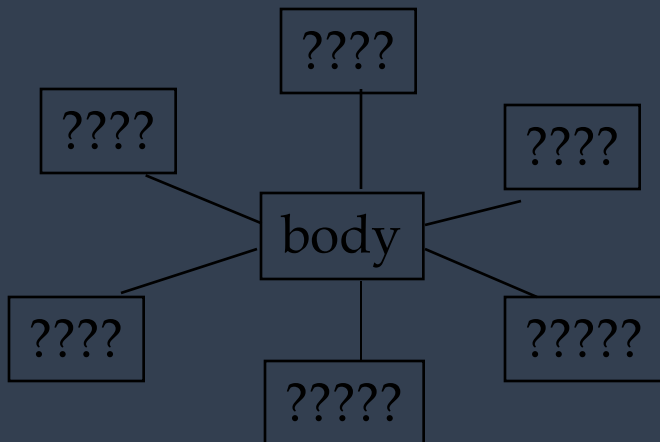


*A working OO system:
a conversation between networks
of objects*

Objects = Attributes, Methods, Associations



สุนัข



Attributes

Model
Company
Color
....

Associations

Brake
LeftFrontWheel
Gear
....

Methods/Operations

Start..
Go..
Stop..
Go Forward
Go Bakcward
Turn Left
Turn Right
Apply Brake
Change Gear
....

*A working OO system:
a conversation between networks
of objects*

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

OO : *Object/Class*

การตั้งชื่อกลุ่ม ที่มี Attribute และ Function เหมือนกันแบบเดียวกัน จัดให้อยู่กลุ่มเดียวกัน เรียกว่าคลาส
อะไรอยู่กลุ่มเดียวกัน เรียกว่าคลาสเดียวกัน

สามเหลี่ยม

- ๕
- ส
- จำนวนพ.ท. ()

สี่เหลี่ยม

- กว้าง
- ยาว
- จำนวนพ.ท.()

วงกลม

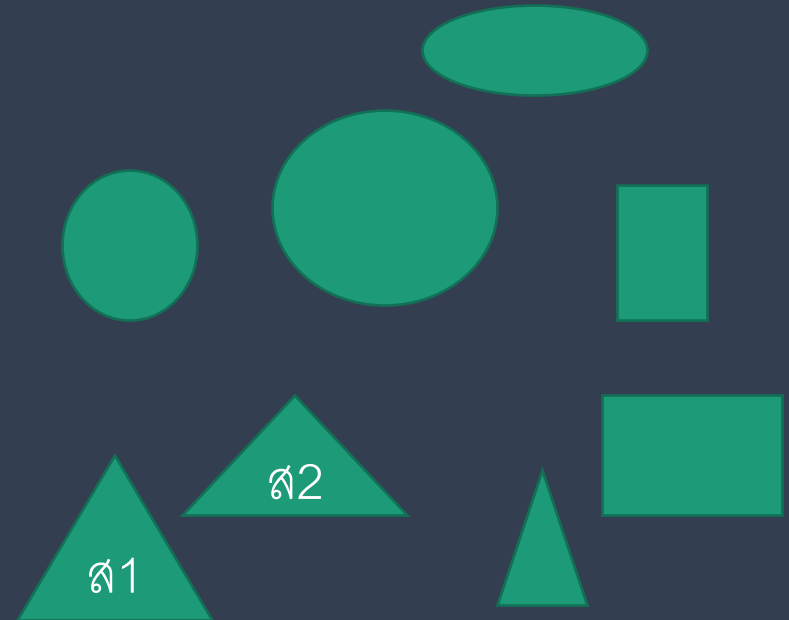
- รัศมี
- จำนวนพ.ท.()

สามเหลี่ยม

- มีมุมสามมุม
- สี่เหลี่ยม
- มีมุมสี่มุม

วงกลม

ไม่มีมุม



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

OO : *Objects Abstraction*

- กำหนด**คุณลักษณะ (Attribute)** ให้กับวัตถุ
 - วัตถุ “Tim” มีคุณลักษณะสำคัญคือ อายุ (25 ปี) เพศ (ชาย) ชื่อ (Tim) เป็นต้น
 - วัตถุ “รถยนต์คันหนึ่ง” สามารถ วิ่ง เลี้ยวซ้าย และเลี้ยวขวาได้ เป็นต้น
- กำหนด**พฤติกรรม/วิธีการกระทำ (Method)** ให้กับวัตถุ
 - วัตถุ “Tim” สามารถ เดิน ยืน นั่ง และนอน ได้ เป็นต้น
 - วัตถุ “รถยนต์คันหนึ่ง” สามารถ วิ่ง เลี้ยวซ้าย และเลี้ยวขวาได้ เป็นต้น
- กำหนด**ความสัมพันธ์/ปฏิสัมพันธ์กับวัตถุอื่น (Association)**
 - วัตถุ “Tim” ประกอบด้วย ลำตัว(body) แขน (arm) 2 แขน ขา(leg) 2 ขา เป็นต้น
 - วัตถุ “รถยนต์คันหนึ่ง” ประกอบด้วย ตัวถัง(body) ล้อ (wheel) 4 ล้อ เป็นต้น
 - วัตถุ “Tim” สามารถสร้างกิจกรรม ให้วัตถุ “รถยนต์คันหนึ่ง” เกิดการเปลี่ยนแปลงสถานะได้ เช่น การขับรถ เป็นต้น

Group of Objects

วัตถุที่มี **Attribute, Method** และ **Association** ของวัตถุหลายๆ ก้อน จะถูกจัดอยู่ในกลุ่มหรือ **คลาส (Class)** เดียวกัน



นายแดง

นายดำ

นายเขียว

Defining Class

- ค่า Attribute, Method และ Association ของวัตถุ จะถูกกำหนดไว้ใน “คลาส” (Class)
- คลาส (Class)
 - เป็นสิ่งที่นำเสนอ Abstract Data Types (ADTs) ของวัตถุ (Object) ที่เราต้องการจำลอง
 - กำหนดรายละเอียดเกี่ยวกับ Attribute, Method และ Association ของวัตถุ (Object)
 - เป็นพิมพ์เขียว (Blueprint) ของวัตถุ (Object)

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

OO : *Objects Abstraction*

Attributes/Properties

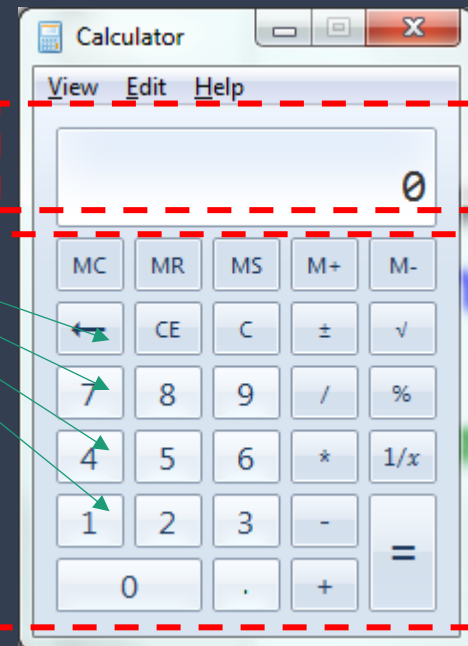
Text
Width
Height
BackColor
ForeColor

button

Function/Method

Click
Enter
เปลี่ยนสีพื้นหลังได้()
เปลี่ยนสีตัวอักษรได้()

Textbox



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

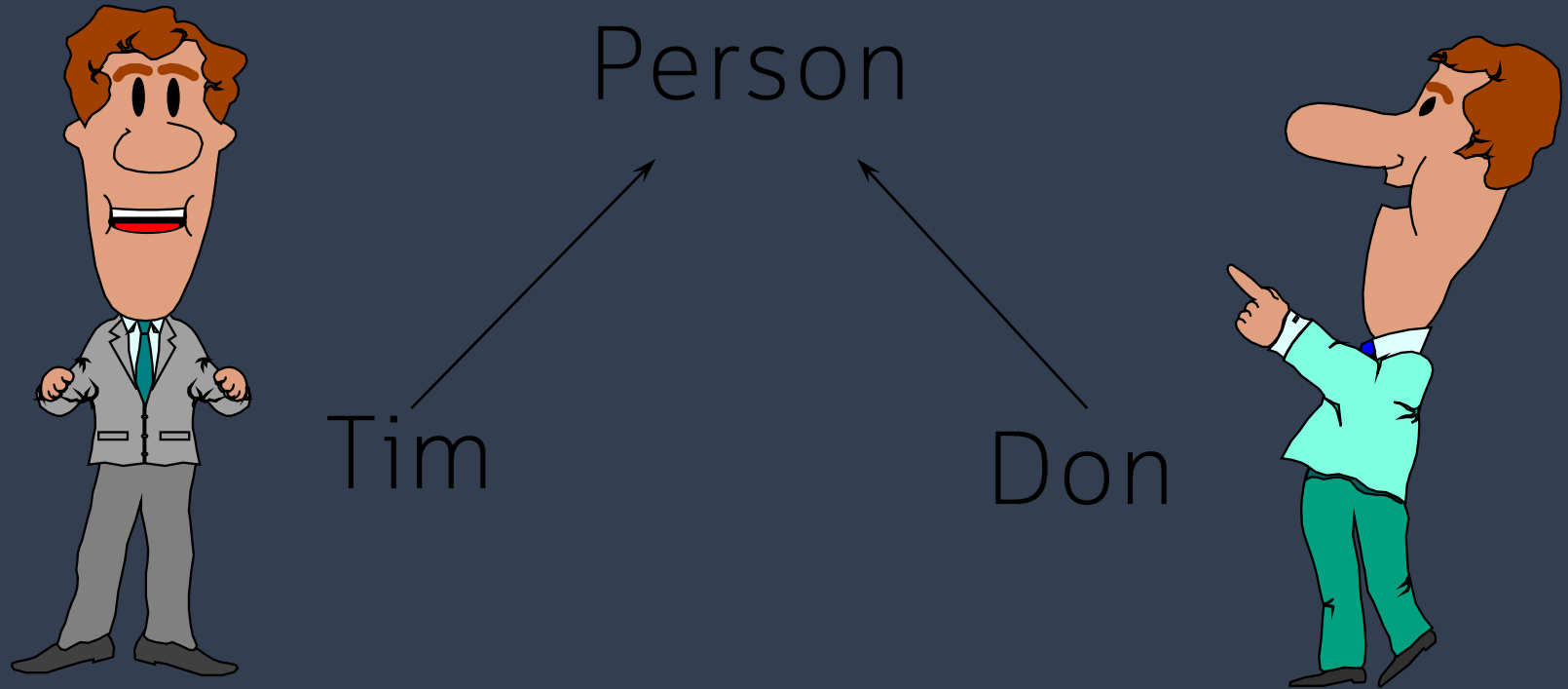
บทที่ 1

OO : Class คืออะไร

คลาส (Class)

- คลาส (Class) เป็นพิมพ์เขียว (Blueprint) ของวัตถุ (Object)
 - ไม่สามารถนำมาใช้ได้โดยตรง
 - กำหนดคุณลักษณะร่วมกันในรูปของ Attribute
 - กำหนดวิธีการกระทำ/พฤติกรรม ร่วมกันของวัตถุ ในรูปของ Method
 - กำหนดความสัมพันธ์ /ปฏิสัมพันธ์กับวัตถุอื่นในรูปของ Association
- เรียกได้ว่า วัตถุ (Object) เป็นสิ่งที่สร้างขึ้น หรือเป็น อินสแตนซ์ (Instance) ของคลาส (Class)

Classes & Instances

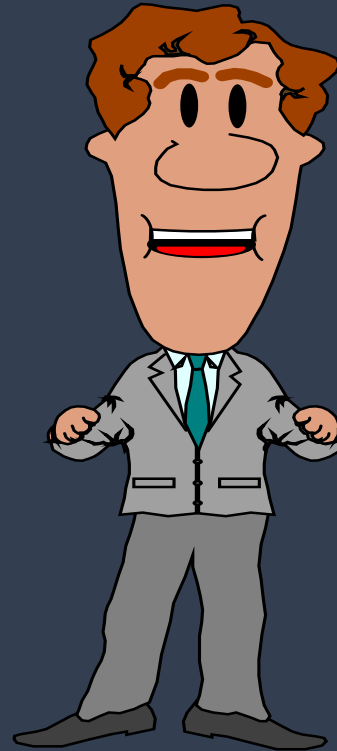


- เรียกได้ว่า คลาส (Class) เป็น พิมพ์เขียว/แบบแปลน (Blueprint) ของ วัตถุ (Object)
- เรียกได้ว่า วัตถุ (Object) เป็น อินสแตนซ์ (Instance) ของคลาส (Class)

Class & Object

PERSON
Age Gender Name myArms myLegs
Jump() Walk() Talk() Sleep() ShakeMyArms()

Class



Object “ນາຍດຳ”

Age = 35
Gender = ສາຍ
Name = ນາຍດຳ
myArms = ແຂນ
myLegs = ທີ

can Jump
can Walk
can Talk
can Sleep
can ShakeMyArms

Class & Object

PERSON
Age Gender Name myArms myLegs
Jump() Walk() Talk() Dance() ShakeMyArms()

Class

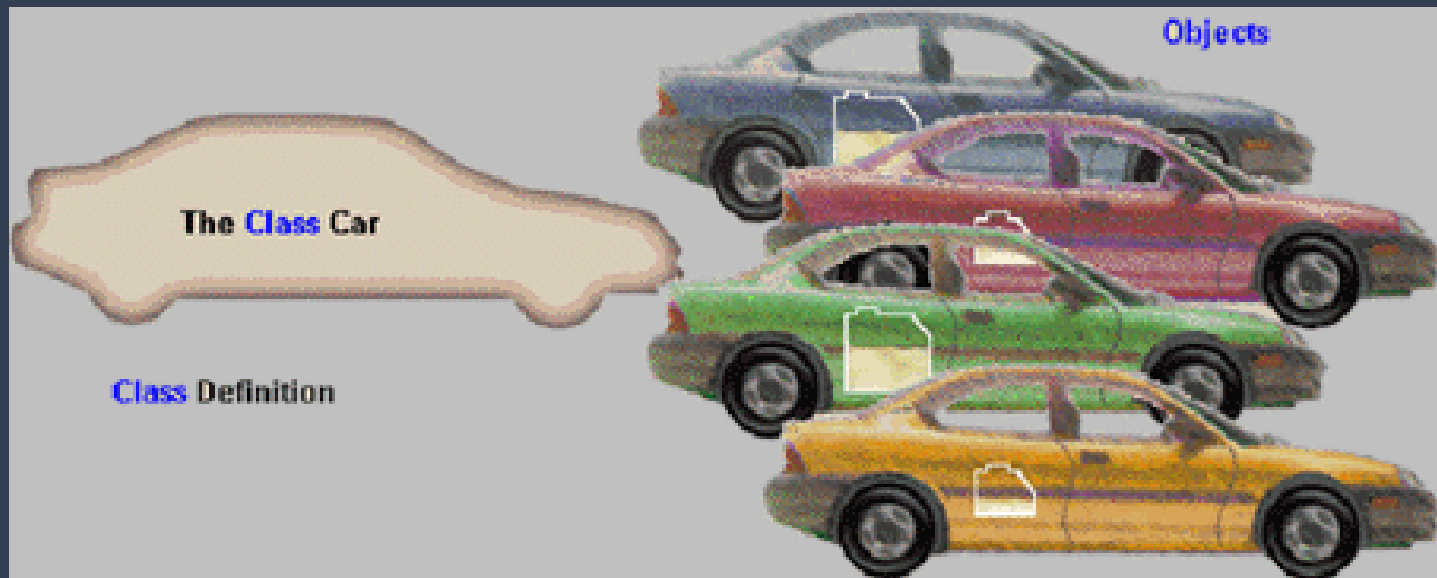


Object “Don”

Age = 25
Gender = Male
Name = Don
myArms = Arms
myLegs = Legs

can Jump
can Walk
can Talk
can Dance
can ShakeMyArms

Classes & Instances



- เรียกได้ว่า **คลาส (Class)** เป็น **พิมพ์เขียว/แบบแปลน (Blueprint)** ของ **วัตถุ (Object)**
- เรียกได้ว่า **วัตถุ (Object)** เป็น **อินสแตนซ์ (Instance)** ของ **คลาส (Class)**

Class & Object

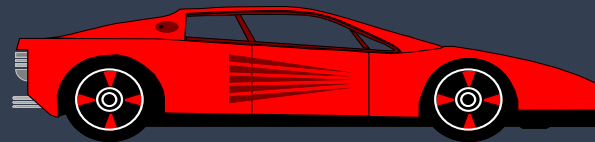
CAR
Model Color Company theGear theBrake
GoForward() GoBackward() TurnLeft() TurnRight() ApplyBrake()

Class



Model = Celica
Color = Red
Company = Toyota
theGear = Gear
theBrake = Brake

can GoForward
can GoBackward
can TurnLeft
can TurnRight
can ApplyBrake



Object “aCar”

Message Passing (1)

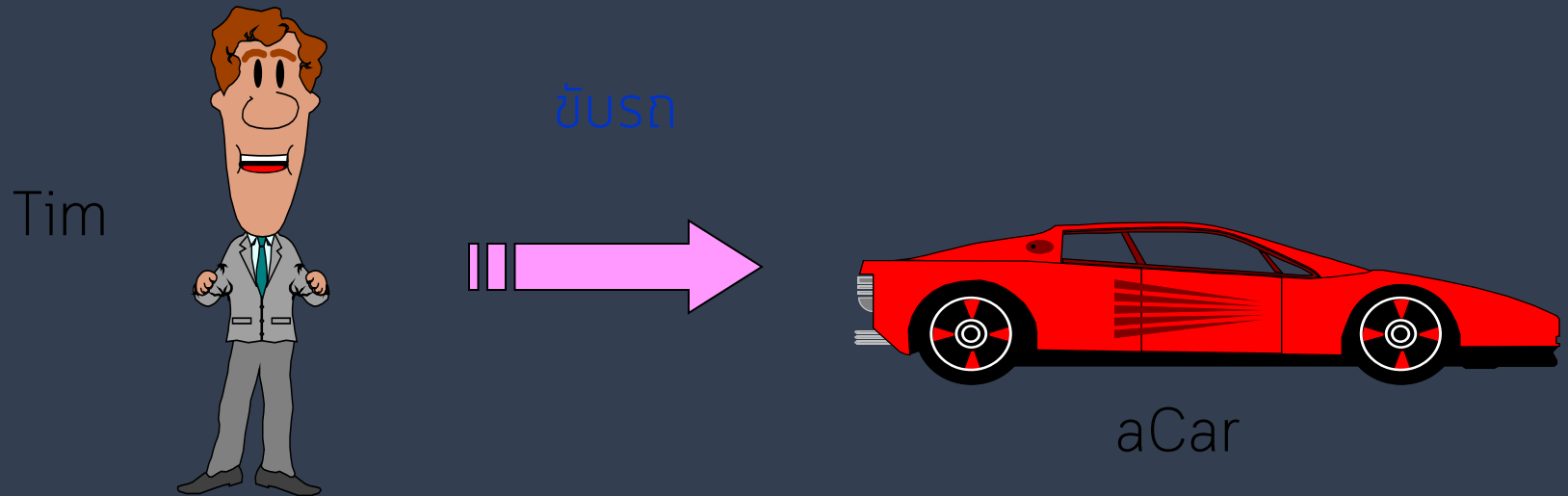
- Message Passing

- คือการที่ วัตถุติดต่อกันด้วยการส่ง ข้อความ (message) ถึงกันและกัน



- มีผลทำให้ วัตถุที่รับข้อความ (received object) เปลี่ยนแปลงสถานะ หรือ กระทำการอย่างใดอย่างหนึ่ง โดยจะทำการประมวลผลตามความต้องการของวัตถุที่ส่งข้อความ หรือ ปฏิเสธคำร้องขอ

Example of Message Passing (1)



ถ้าวัตถุ Tim ต้องการขับรถ aCar จะต้องทำอย่างไร?

Message passing (2)

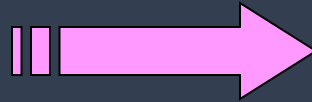
- การส่ง Message ไปยังวัตถุใดๆ แล้ววัตถุนั้นกระทำการอย่างใดอย่างหนึ่ง
 - เป็นการเรียก Method ที่ถูกกำหนดไว้ในวัตถุนั้นมาทำงาน
 - เปรียบเทียบได้กับการเรียก function call หรือ procedure call ที่มีใน Procedural Programming โดยผ่าน Interface
 - แต่ละวัตถุ ต้องรับผิดชอบ พฤติกรรมของตนเอง โดยผ่านการเรียกใช้ Method

Example of Message Passing (2)

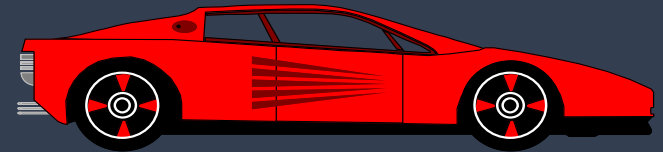
ถ้าวัตถุ Tim ต้องการขับรถ
aCar



Tim

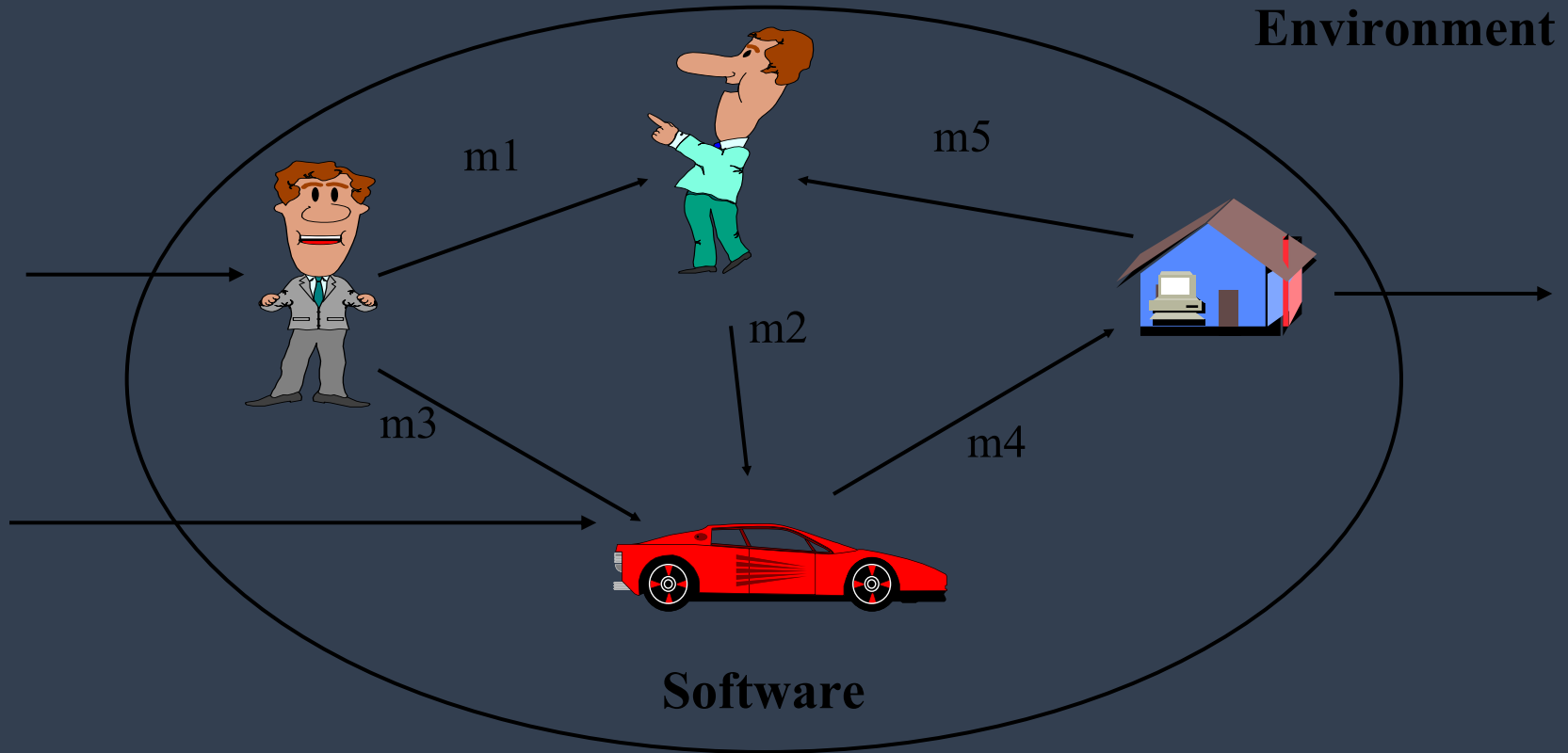


```
aCar.GoForward(100)  
aCar.goBackward()  
aCar.TurnLeft()  
aCar.TurnRight()  
aCar.ApplyBrake()
```



aCar

Object-Oriented Software



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

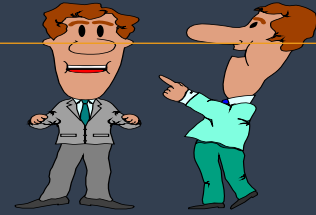
บทที่ 1

OO : *Characteristics of an Object*

Characteristics of an Object

- Identity
- Information Hiding
- Encapsulation
- Inheritance
- Polymorphism

Identity

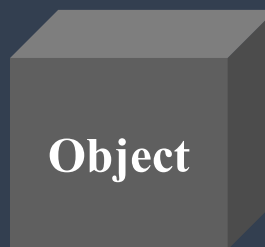


- วัตถุ (Object) จะต้องมัลักษณะเฉพาะประจำตัว (Identity)
 - วัตถุใดๆ สามารถมีคุณลักษณะ (Attribute) เหมือนๆ กันได้ เช่น วัตถุ “Tim” และ “Don” เป็นเพศชาย และอายุ 25 ปี เหมือนกัน
 - วัตถุใดๆ สามารถมีพฤติกรรม (Behavior/Method) เหมือนๆ กันได้ เช่น วัตถุ “Tim” และ “Don” สามารถกิน เดิน นอน นั่ง ได้เหมือนกัน
 - วัตถุใดๆ สามารถความสัมพันธ์เกี่ยวข้อง (Association) กับวัตถุอื่นๆ เหมือนๆ กันได้ เช่น วัตถุ “Tim” และ “Don” ประกอบด้วย 2 แขน และ 2 ขา เหมือนๆ กัน
 - วัตถุ “Tim” และ “Don” สร้างจากคลาส (Class) เดียวกันคือ “Person”
- แต่เราถือว่า วัตถุ “Tim” และวัตถุ “Don” เป็นคนละวัตถุกัน

Encapsulation

- พิจารณาวัตถุ (Object) ได้เสมือน กล่องดำ (Black Box)
 - วัตถุใด ต้อง “รู้” วิธีหรือขั้นตอนการทำงานภายใน (Method) ของตนเอง
 - วัตถุอื่น ไม่จำเป็นต้องรู้ว่าวิธี หรือขั้นตอนการทำงานภายในเป็นอย่างไร
 - วัตถุอื่นต้องการเรียกใช้ Method ของวัตถุนี้ได้ โดยส่ง Message มา โดยที่ Message จะต้องอยู่ในรูปแบบที่ตกลงกันไว้ก่อน

IncomingMessage



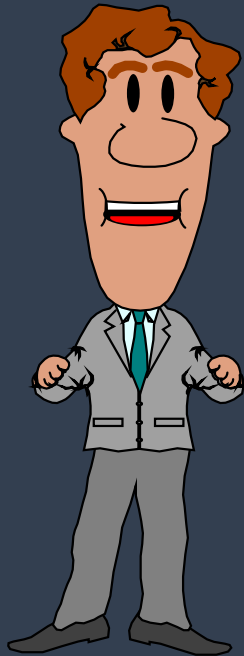
black box

Outgoing Message



Example of Encapsulation

นายดำ



Tim.jump()
Tim.walk()
Tim.talk()
Tim.dance()
Tim.shakeMyArms()

Information Hiding

■ พิจารณาวัตถุ (Object)

- สามารถกำหนดขอบเขตการเข้าถึง Attribute และ Method
- กำหนดว่าใครสามารถเห็นหรือใช้ Attribute และ Method ของวัตถุนั้นได้บ้าง



Object Diagram A.



Object Diagram B.



Object Diagram C.

Information Hiding and Encapsulation

PERSON
Age
Gender
Name
myArms
myLegs
Jump()
Walk()
Talk()
Dance()
ShakeMyArms()
getAge()

- Q: จะคำนวณค่า "Age" จาก PERSON ได้อย่างไร?
 - เก็บค่า วัน-เดือน-ปีเกิด ไว้ แล้วคำนวณค่า "Age" เมื่อมี request?
 - เก็บค่าอายุปัจจุบัน (current age) และ update ทุกๆ ปี?
- A: don't care!!!
 - ตราบใดที่เราสามารถ maintain "getAge()" interface

Advantages of Information Hiding and Encapsulation

■ Maintainability

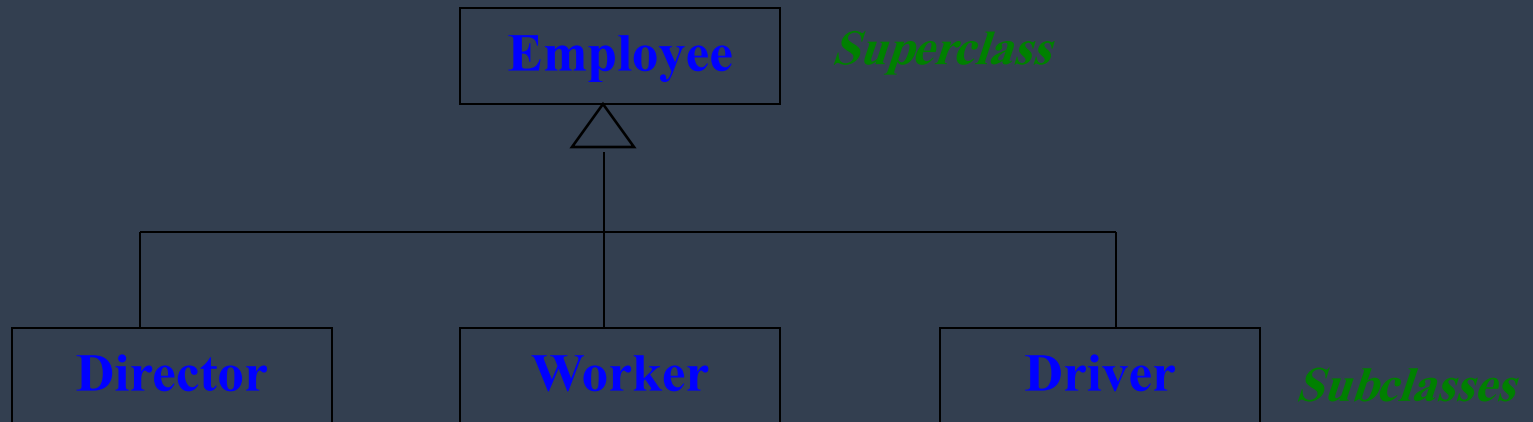
- เป็นการพิจารณาโครงสร้างซอฟต์แวร์ในระดับนามธรรม (Abstraction) หรือในระดับสูง (High-Level) โดยไม่คำนึงถึงรายละเอียด วิธีการ หรือภาษาที่จะนำไปพัฒนา (Implementation)

■ Flexibility

- ทำให้การพัฒนาซอฟต์แวร์เชิงวัตถุมีความยืดหยุ่น (Flexibility) สามารถเปลี่ยนแปลงได้ง่าย (Ease of Change)
- ระบบ Software ใดๆ สามารถปฏิบัติงานร่วมกันได้อย่างสะดวก และมีประสิทธิภาพ เนื่องจากไม่จำเป็นต้อง “รู้” รายละเอียดการทำงานภายในของระบบ

Inheritance (1)

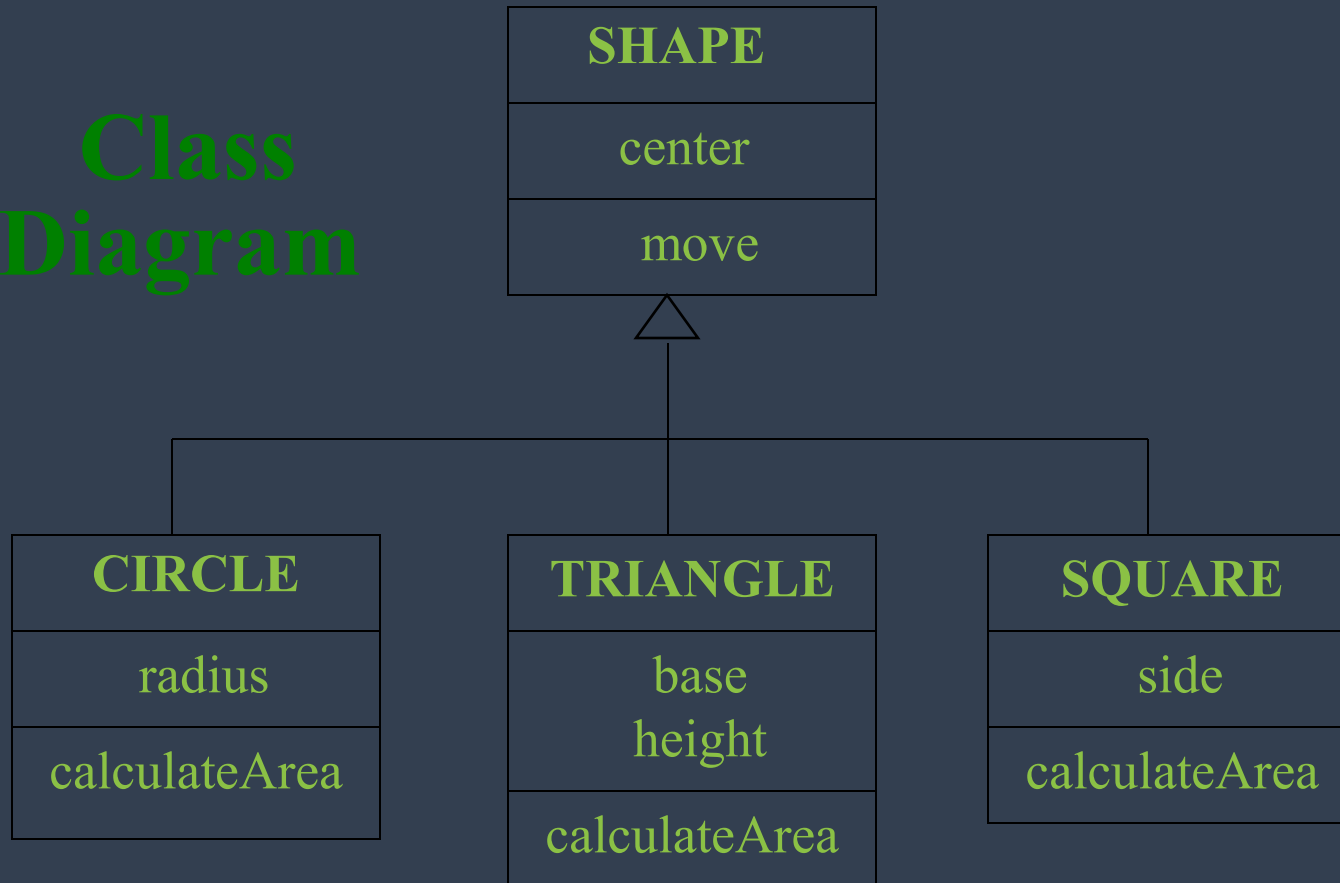
- การถ่ายทอดคุณสมบัติ (Inheritance)
 - คือการที่ class ที่ต่างกันมี Attributes และ Methods ที่เหมือนกัน



- **Subclasses** รับถ่ายทอดคุณสมบัติ (inherit) ของ properties จาก **Superclass**

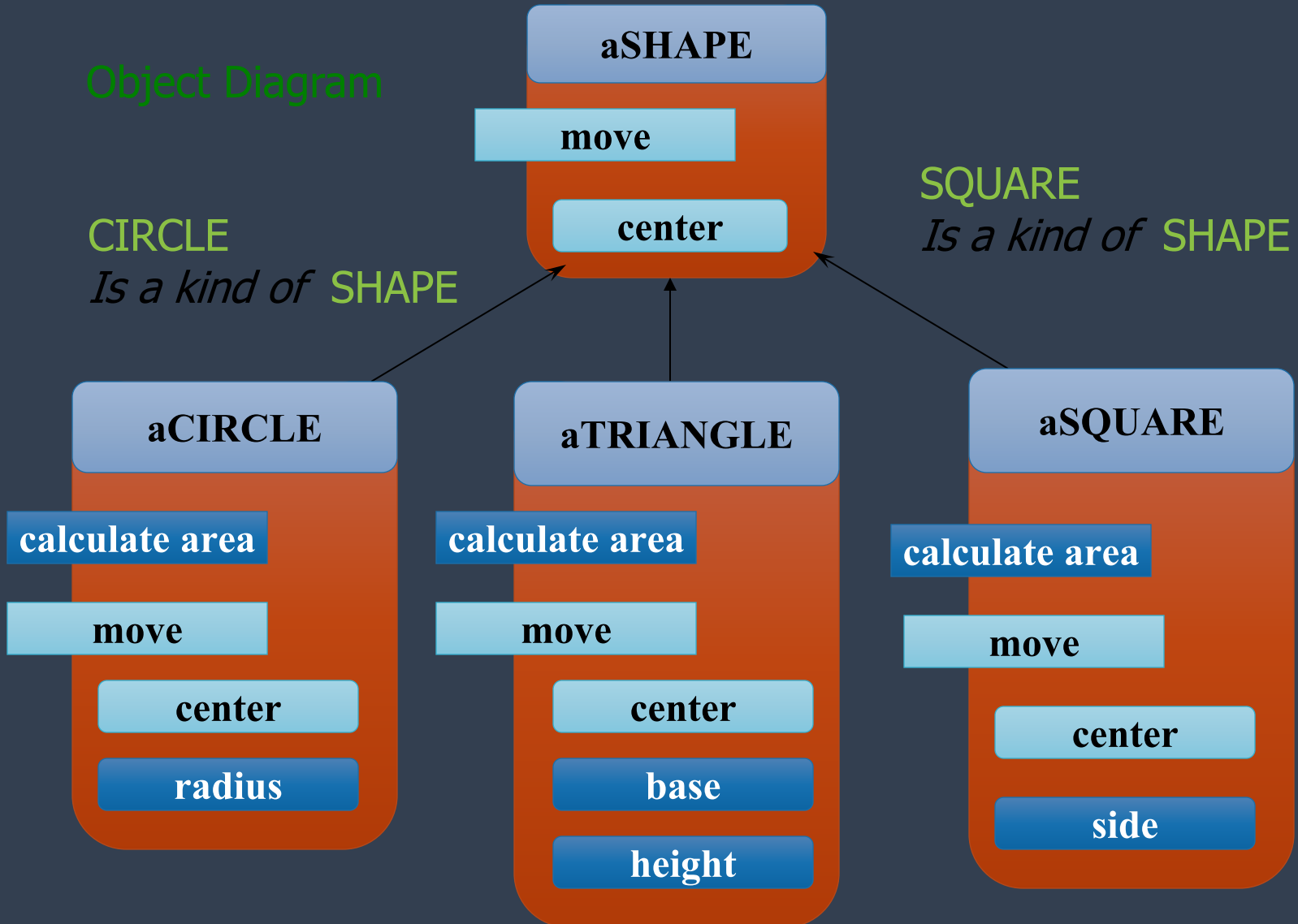
Inheritance (2)

Class Diagram



Inheritance (2)

Object Diagram



Generalization and Specialization

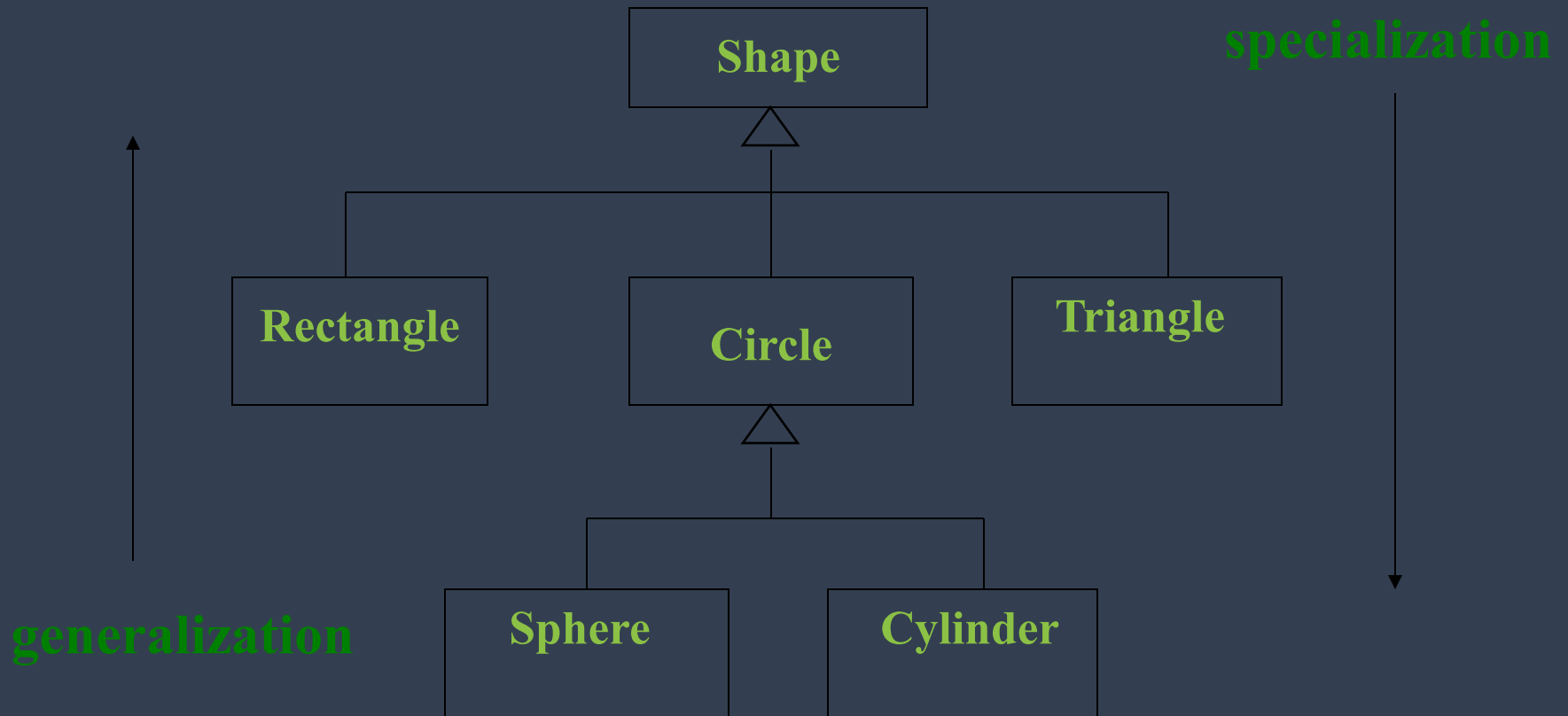
■ Generalization

- การกำหนดคุณลักษณะร่วมกันจากคลาสหลายๆ คลาส และนำมากำหนดไว้ในคลาสที่อยู่ในลำดับชั้นสูงขึ้น (superclass) ของโครงสร้างลำดับชั้น (Class hierarchy)
- bottom-up

■ Specialization

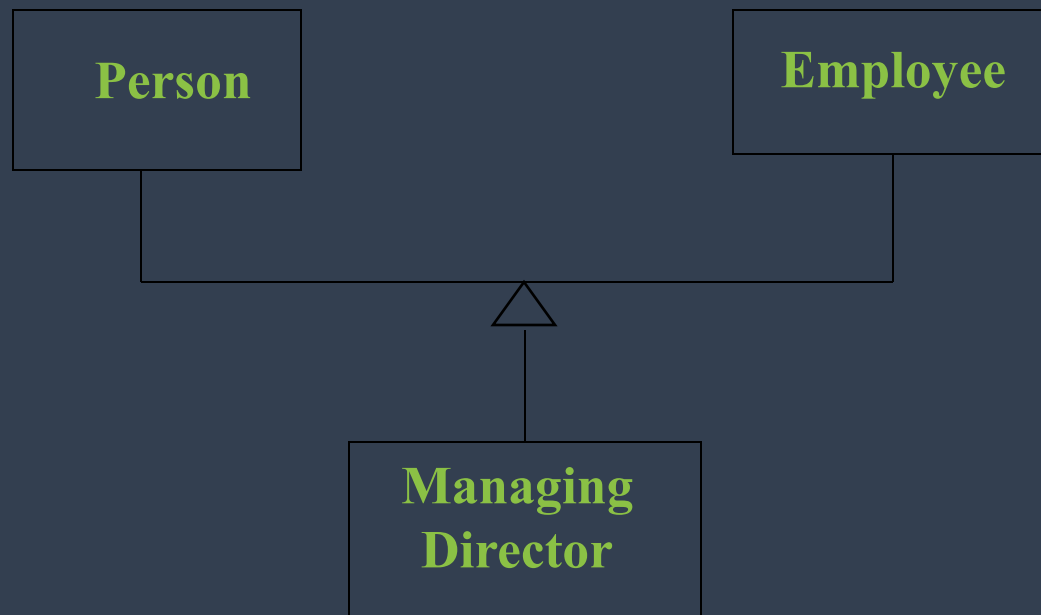
- การกำหนดคลาสใหม่ขึ้น (Subclass) โดยรับถ่ายทอดคุณสมบัติ (inherit) คุณสมบัติทั้งหมดอันได้แก่ คุณลักษณะและพฤติกรรม จากคลาสเดิม
- top-down

Inheritance (3)



Multiple Inheritance

- คลาส 1 คลาส รับถ่ายทอดคุณสมบัติ (inherit) จาก superclass ตั้งแต่ 2 คลาสขึ้นไป



Advantages of Inheritance

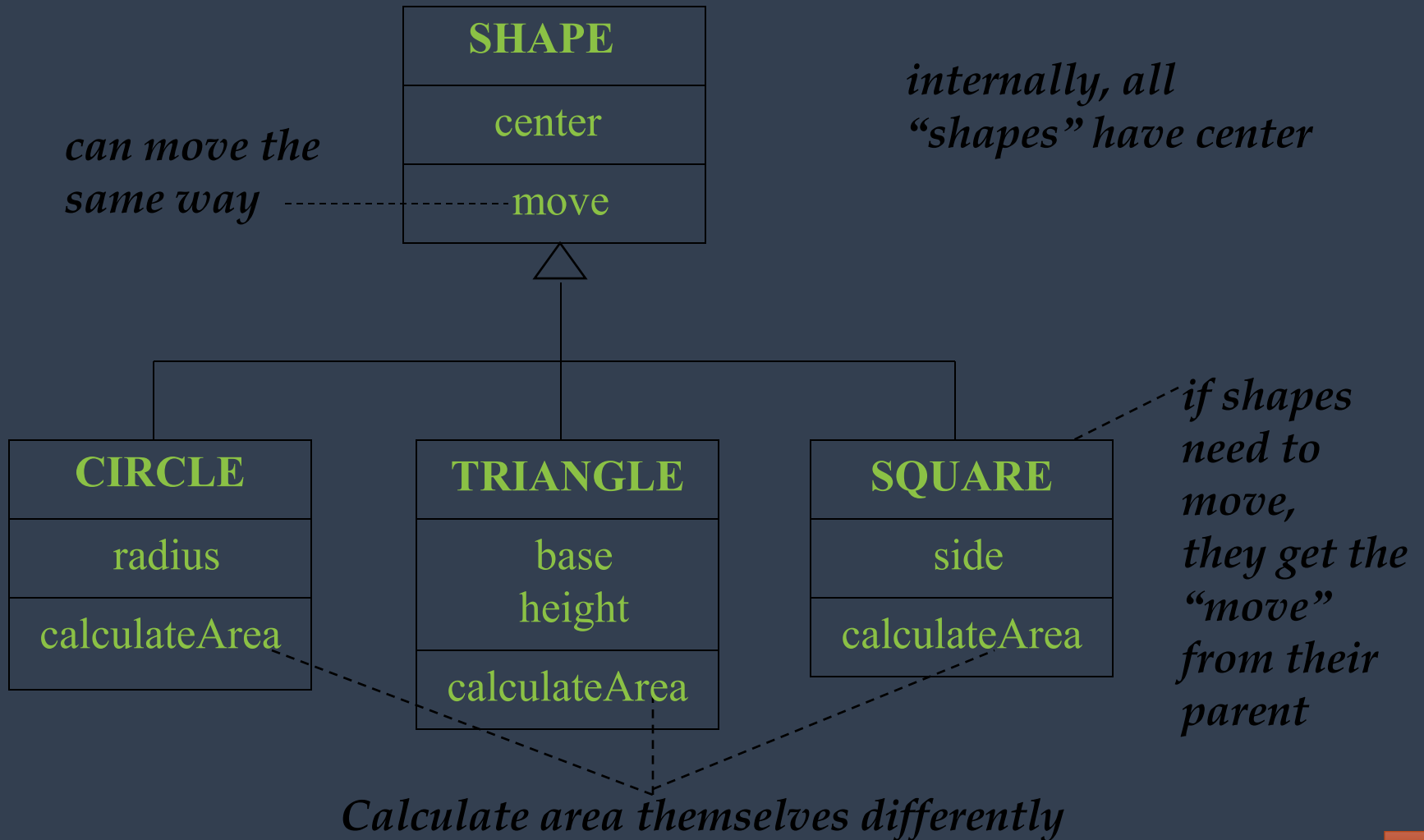
- Avoiding redundancies
 - Source Code ในส่วนที่เหมือนกันจะถูกเขียนขึ้นเพียงครั้งเดียว
- Code reuse
 - สามารถนำไปใช้ใหม่ได้ (reuse) ได้โดยการรับถ่ายทอดคุณสมบัติ
- Reduced code size and complexity
 - เนื่องจาก Source Code ถูกเขียนขึ้นเพียงครั้งเดียว และถูกนำไปใช้ใหม่ได้ไม่จำกัด จึงเป็นการประหยัดเนื้อที่ Source Code และลดความซับซ้อนของ Source Code

Polymorphism

- Polymorphism วัตถุต่าง ๆ ที่มีพฤติกรรมแบบเดียวกันอยู่ในตัว แต่รายละเอียดวิธีการกระทำอาจแตกต่างกันได้ ขึ้นอยู่กับว่าวัตถุนั้นถูกนำไปใช้แบบใด
 - เช่น วัตถุ “SQUARE” และวัตถุ “CIRCLE” ต่างก็สามารถถูก “calculateArea” ได้ แต่วิธีการ “calculateArea” ของทั้ง 2 วัตถุ อาจไม่เหมือนกัน
- วัตถุต่าง ๆ มีวิธีคุณลักษณะ และรายละเอียดวิธีการกระทำที่แตกต่างกัน แต่มองให้เหมือนกันได้ จากคุณลักษณะและพฤติกรรมที่วัตถุเหล่านั้นมีร่วมกัน
 - เช่น วัตถุ “SQUARE” และวัตถุ “CIRCLE” มีวิธีการ “calculate Area” ต่างกัน แต่สามารถมองคุณสมบัติร่วมกันอันได้แก่ ทั้ง 2 วัตถุ ต่างก็เป็น วัตถุ “SHAPE” ที่สามารถ “calculate Area” ได้



Polymorphism



1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น

(Concept of Object-Oriented Technology)

บทที่ 1

OO : Advantages of Polymorphism

Advantages of Polymorphism

▪ Software Interoperability

ระบบ Software ใดๆ สามารถปฏิบัติงานร่วมกันได้อย่างสะดวก และมีประสิทธิภาพ เนื่องจากสามารถมองภาพรวมของวัตถุ ที่อยู่ภายในระบบเป็นรูปแบบที่มีลักษณะร่วมกัน โดยไม่จำเป็นต้อง “รู้” รายละเอียดการทำงานภายในของวัตถุ

Why Object Orientation?

■ Key Benefits of Object Orientation

- นำเสนอมุมมอง และแนวทางที่เป็นประโยชน์ในการจัดการกับปัญหาความซับซ้อนของ Software ได้ดีกว่า Traditional Techniques
- สนับสนุน การติดต่อสื่อสารที่มีประสิทธิภาพในช่วงวงจรการพัฒนา Software (Software Development Life Cycle)
- ตอบสนองต่อ Software Crisis
- ลดผลกระทบอันเกิดจากการเปลี่ยนแปลง (Change) ของ Software
- สนับสนุนการนำ components กลับมาใช้ใหม่ (High Level reuse of Components)
- สนับสนุนการแก้ไข Problems of Software อันได้แก่ Reusability, Extendability, Interoperability เป็นต้น

What do we learn about OOT?

- OO Terms and Concepts
 - Object and Class
 - Inheritance, Encapsulation and Polymorphism
 - Relationship and Interaction between classes
- OO Analysis and Design
 - Object Modeling with UML
- OO Programming Languages
 - Java Language
- Distributed Object Technology
 - RMI and CORBA
 - .NET

1

บทที่ 1

หลักการเชิงวัตถุเบื้องต้น (Concept of Object-Oriented Technology)

บทที่ 1

OO : Summary

- **Software, Systems and Software Systems**
- **Software Crisis**
- **Traditional Software Development**
- **Object-Oriented Software Development**
- **Principle of Object Orientation**
- **Key Benefits of Object-Oriented Software Development**
- **What do we learn in this course?**