



## แผนการจัดการเรียนรู่มุ่งเน้นสมรรถนะ

ชื่อวิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ

รหัสวิชา 31901-2006 ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง

ประเภทวิชา ประเภทวิชาอุตสาหกรรมดิจิทัลและเทคโนโลยีสารสนเทศ

สาขาวิชา เทคโนโลยีสารสนเทศ

จัดทำโดย

นางสุปรียา รัตนวิทยาพันธ์ุ์

วิทยาลัยเทคนิคชลบุรี

สำนักงานคณะกรรมการการอาชีวศึกษา

กระทรวงศึกษาธิการ



## หลักสูตรรายวิชา

ชื่อวิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006 ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

หลักสูตรประกาศนียบัตรวิชาชีพ

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง

สาขาวิชา เทคโนโลยีสารสนเทศ

### อ้างอิงมาตรฐาน

–

### ผลลัพธ์การเรียนรู้ระดับรายวิชา

พัฒนาซอฟต์แวร์ตามหลักการเชิงวัตถุด้วยความรับผิดชอบ ความละเอียด รอบคอบ และถูกต้อง

จุดประสงค์รายวิชา เพื่อให้

1. เข้าใจเกี่ยวกับหลักการพัฒนาซอฟต์แวร์เชิงวัตถุ
2. มีทักษะในการพัฒนาซอฟต์แวร์เชิงวัตถุ
3. มีเจตคติและกิจนิสัยที่ดีในการปฏิบัติงานด้วยความรับผิดชอบ ความละเอียด รอบคอบ และถูกต้อง
4. มีความสามารถประยุกต์ใช้ภาษาคอมพิวเตอร์เชิงวัตถุเพื่อพัฒนาซอฟต์แวร์

### สมรรถนะรายวิชา

1. ประมวลความรู้เกี่ยวกับการพัฒนาซอฟต์แวร์ตามหลักการเชิงวัตถุ
2. พัฒนาซอฟต์แวร์ด้วยภาษาคอมพิวเตอร์เชิงวัตถุ
3. ประยุกต์ใช้ภาษาคอมพิวเตอร์เชิงวัตถุเพื่อพัฒนาซอฟต์แวร์

### คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการพื้นฐานของการโปรแกรมเชิงวัตถุด้วยภาษาคอมพิวเตอร์เชิงวัตถุ การติดตั้งและกำหนดการทำงานของโปรแกรมที่เกี่ยวข้องกับการพัฒนาโปรแกรมเชิงวัตถุ ลำดับขั้นตอนการ เขียนโปรแกรมเชิงวัตถุ การสร้างและเรียกใช้งาน Classes, Objects, Attributes, Methods, Constructors, Modifiers, Encapsulation, Packages, Inheritance, Polymorphism, Abstraction, Interface, Exceptions Handlings, Threads และเชื่อมต่อฐานข้อมูลในการพัฒนาซอฟต์แวร์เชิงวัตถุในงานอาชีพ

## หน่วยการเรียนรู้

หน่วยที่	ชื่อหน่วยการเรียนรู้	จำนวน ชั่วโมง	สัปดาห์ที่
1	หลักการพื้นฐานของการโปรแกรมเชิงวัตถุ	5	1
2	การติดตั้งและกำหนดการทำงานของโปรแกรม	5	2
3	Classes และ Objects	10	3-4
4	Constructors และ Modifiers	5	5
5	Encapsulation และ Packages	10	6-7
6	Inheritance และ Polymorphism	10	8-9
7	Abstraction และ Interface	10	10-11
8	Exception Handling และ Threads	10	12-13
9	การเชื่อมต่อฐานข้อมูลและการประยุกต์ใช้ในงานอาชีพ	5	14
10	สอบปลายภาค	5	15

หน่วยการเรียนรู้และสมรรถนะประจำหน่วย

ชื่อหน่วย	สมรรถนะ		
	ความรู้	ทักษะ	คุณลักษณะที่พึงประสงค์
หน่วยที่ 1 หลักการพื้นฐานของการโปรแกรมเชิงวัตถุ	เข้าใจหลักการพื้นฐานของการโปรแกรมเชิงวัตถุ	อธิบายหลักการพื้นฐานของการโปรแกรมเชิงวัตถุ	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 2 การติดตั้งและกำหนดการทำงานของโปรแกรม	เข้าใจหลักการการติดตั้งและกำหนดการทำงานของโปรแกรม	ติดตั้งและกำหนดการทำงานของโปรแกรม  -การทดสอบและแก้ไข	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 3 Classes และ Objects	เข้าใจหลักการ Classes และ Objects	-เขียนโปรแกรมสร้าง Classes และ Objects  -การทดสอบแก้ไขและเรียกใช้	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 4 Constructors และ Modifiers	เข้าใจหลักการ Constructors และ Modifiers	เขียนโปรแกรมสร้าง Constructors และ Modifiers  -การทดสอบแก้ไขและเรียกใช้	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 5 Encapsulation และ Packages	เข้าใจหลักการ Encapsulation และ Packages	เขียนโปรแกรมสร้าง Encapsulation และ Packages  -การทดสอบแก้ไขและเรียกใช้	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน

หน่วยที่ 6 Inheritance และ Polymorphism	เข้าใจหลักการ Inheritance และ Polymorphism	เขียนโปรแกรมสร้าง Inheritance และ Polymorphism -การทดสอบแก้ไขและเรียกใช้	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 7 Abstraction และ Interface	เข้าใจหลักการ Abstraction และ Interface	เขียนโปรแกรมสร้าง Abstraction และ Interface -การทดสอบแก้ไขและเรียกใช้	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 8 Exception Handling และ Threads	เข้าใจหลักการภาษา Exception Handling และ Threads	เขียนโปรแกรมสร้าง Exception Handling และ Threads -การทดสอบแก้ไขและเรียกใช้	ทำงานเป็นระเบียบ ตรงต่อเวลา ช่วยเหลือเพื่อน
หน่วยที่ 9 การเชื่อมต่อฐานข้อมูลและการประยุกต์ใช้ในงานอาชีพ	เข้าใจหลักการของการเชื่อมต่อฐานข้อมูล	เขียนโปรแกรม เชื่อมต่อฐานข้อมูล -การจัดการข้อผิดพลาด -การออกแบบฐานข้อมูล -การทดสอบและการปรับปรุง	



## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 1

ชื่อหน่วย หลักการพื้นฐานของการโปรแกรมเชิงวัตถุ

สอนครั้งที่ 1

ชั่วโมงรวม 5

### 1. สาระสำคัญ

หลักการพื้นฐานของการโปรแกรมเชิงวัตถุ (Object-Oriented Programming - OOP)

ประกอบด้วยสี่แนวคิดหลัก ได้แก่:

1. **Encapsulation (การห่อหุ้ม):** การซ่อนรายละเอียดของข้อมูลภายใน object ไม่ให้สามารถเข้าถึงจากภายนอกได้โดยตรง
  2. **Abstraction (การทำให้เป็นนามธรรม):** การซ่อนความซับซ้อนของข้อมูล และแสดงเฉพาะส่วนที่จำเป็นต่อผู้ใช้
  3. **Inheritance (การสืบทอด):** คลาสสามารถรับคุณสมบัติจากคลาสอื่น (คลาสแม่) ได้
  4. **Polymorphism (การพ้องรูป):** ความสามารถของ object ที่สามารถถูกใช้งานได้หลายรูปแบบ
- เหล่านี้เป็นหลักการพื้นฐานที่ช่วยให้การพัฒนาโปรแกรมเป็นไปได้อย่างมีระเบียบและสามารถจัดการกับความซับซ้อนของโปรแกรมได้ดี

### 2. สมรรถนะประจำหน่วย

- 2.1 อธิบายหลักการพื้นฐานของการโปรแกรมเชิงวัตถุ
- 2.2 ออกแบบโปรแกรมโดยใช้แนวคิดเชิงวัตถุเพื่อแก้ปัญหาต่าง ๆ ได้

### 3. จุดประสงค์การเรียนรู้

- 3.1 ดานความรู้
  - 3.1.1 เข้าใจหลักการพื้นฐานของการโปรแกรมเชิงวัตถุ
  - 3.1.2 อธิบายหลักการพื้นฐานของ OOP
- 3.2 ดานทักษะ
  - 3.2.1 สามารถอธิบายหลักการพื้นฐานของ OOP
  - 3.2.2 สามารถออกแบบและพัฒนาโปรแกรมโดยใช้แนวคิด OOP ในการแก้ไขปัญหาต่าง ๆ
- 3.3 คุณลักษณะที่พึงประสงค์
  - 3.3.1 การตรงต่อเวลา
  - 3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4. เนื้อหาสาระการเรียนรู้

- 4.1 การเขียนโปรแกรมเชิงวัตถุ
- 4.2 เปรียบเทียบแนวคิดระหว่างการเขียนโปรแกรมเชิงกระบวนการ และเชิงวัตถุ

#### 4.3 หลักการพื้นฐานของการโปรแกรมเชิงวัตถุ

##### 5.1 การนำเขาสู่บทเรียน

1. เป็นชั่วโมงแรกในการเรียน การพัฒนาซอฟต์แวร์เชิงวัตถุ ที่ครูผู้สอนและนักเรียนได้พบกันเป็นครั้งแรก โดยครูได้แนะนำตัวเอง พร้อมกับให้นักเรียนทุกคนทำการแนะนำตัวเอง เพื่อ เป็นการสร้างความคุ้นเคยก่อนการเรียน หลังจากนั้น ครูได้ชี้แจงรายละเอียดเกี่ยวกับ การเรียนให้นักเรียนทุกคนในชั้นเรียน ได้ทราบ และยึดถือเป็นข้อปฏิบัติในการเรียนวิชาดังกล่าว เช่น การปฏิบัติตนในขณะที่อยู่ในห้อง เรียน เวลาเรียน การขาดเรียน การส่งงาน การเก็บคะแนนระหว่างภาคเรียน การประเมิน ผลปลายภาคเรียน และการลงปฏิบัติงาน โดยครูจะแจ้งรายละเอียดให้นักเรียนทุกคน รับทราบ เพื่อเป็นแนวทางในการปฏิบัติ

2. ครูสอบถามมีนักเรียนคนใดไม่เข้าใจ หากมีข้อสงสัย หรือคิดว่าปฏิบัติตามระเบียบที่ครูชี้แจงไม่ได้ ให้สอบถามครูได้ เพราะหลังจากเริ่มทำการสอนแล้ว จะไม่มีการชี้แจงรายละเอียดดังกล่าวอีก

3. ครูแจ้งวัตถุประสงค์สมรรถนะรายวิชาและคำอธิบายรายวิชา ให้นักเรียนได้รับทราบ

4. ครูนำเข้าสู่บทเรียนโดยถามนักเรียนเกี่ยวกับ เรื่องการพัฒนาซอฟต์แวร์เชิงวัตถุ

5. ครูกล่าวถึงการพัฒนาซอฟต์แวร์เชิงวัตถุ

##### 5.2 การเรียนรู

ครูอธิบาย พร้อมยกตัวอย่าง และใช้เอกสารประกอบการเรียน เนื้อหาเรื่อง การพัฒนาซอฟต์แวร์เชิงวัตถุ

##### 5.3 การสรุป

ครูสรุปเนื้อหาเรื่อง อธิบายข้อสงสัย ตอบคำถาม

##### 5.4 การวัดและประเมินผล

1. ประเมินผลจากใบงาน
2. แบบทดสอบ
3. สังเกต

#### 6.สื่อการเรียนรู/แหล่งการเรียนรู

##### 6.1 สื่อสิ่งพิมพ์

.....

.....

.....

##### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

6.3 หุนจำลองหรือของจริง (ถ้ามี)

.....  
.....

6.4 อื่นๆ (ถ้ามี)

.....  
.....

7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....  
.....  
.....

8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....  
.....





## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 2

ชื่อหน่วย การติดตั้งและกำหนดการทำงานของโปรแกรม

สอนครั้งที่ 2

ชั่วโมงรวม 5

### 1. สารสำคัญ

ภาษาจาวาเป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุโดยมีจุดเด่นที่สามารถทำงานได้กับระบบคอมพิวเตอร์หลายแพลตฟอร์ม ปี ค.ศ. 2010 บริษัท ออราเคิลได้ซื้อบริษัท ซันไมโครซิสเต็มส์ ทำให้การพัฒนาเทคโนโลยีภาษาจาวาขึ้นกับทีมงานของ บริษัทออราเคิล

เทคโนโลยีภาษาจาวาประกอบไปด้วยองค์ประกอบหลักๆ 3 ส่วนคือ

1. Java Virtual Machine (JVM) ทำหน้าที่แปลงโปรแกรมไบท์โค้ด (Bytecode) ที่ผ่านการคอมไพล์จากไฟล์ภาษาจาวาให้เป็นภาษาเครื่อง เพื่อทำงานตามโปรแกรม

2. Java Runtime Environment (JRE) ใช้ในการรันโปรแกรมภาษาจาวาที่ผ่านการคอมไพล์จากไฟล์ภาษาจาวาซึ่งประกอบไปด้วย JVM และ Application Programming Interface (API) ที่ได้รวบรวมคลาสและอินเตอร์เฟซต่างๆ ที่จำเป็นต่อการใช้งานโปรแกรมภาษาจาวา

3. Java Development Kit (JDK) เป็นชุดพัฒนาโปรแกรมด้วยภาษาจาวาซึ่งมีเครื่องมือช่วยสำหรับนักพัฒนาโปรแกรม เช่น คอมไพเลอร์ ดีบักเกอร์ รันไทม์ ไลบรารี และโปรแกรมสนับสนุนอื่นๆ

### 2. สมรรถนะประจำหน่วย

- 2.1 ติดตั้ง โปรแกรมภาษาจาวาได้
- 2.2 ติดตั้ง โปรแกรม Edit Plus ได้
- 2.3 กำหนดค่าในการติดตั้งโปรแกรมได้
- 2.4 คอมไพล์และรันโปรแกรมได้

### 3. จุดประสงค์การเรียนรู้

- 3.1 ดานความรู้
  - 3.1.1 เข้าใจขั้นตอนการติดตั้งโปรแกรม
  - 3.1.2 อธิบายขั้นตอนการติดตั้งโปรแกรม
- 3.2 ดานทักษะ
  - 3.2.1 ติดตั้งโปรแกรมภาษาจาวาได้
  - 3.2.2 กำหนดค่าเริ่มต้นให้กับโปรแกรมได้
  - 3.2.3 คอมไพล์และรันโปรแกรม
- 3.3 คุณลักษณะที่พึงประสงค์
  - 3.3.1 การตรงต่อเวลา

3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

#### 4. เนื้อหาสาระการเรียนรู้

- 4.1 การติดตั้ง JDK (Java Development Kit)
- 4.2 การกำหนดค่า กำหนดค่า Environment Variables
- 4.3 การติดตั้ง โปรแกรม Edit Plus
- 4.4 การกำหนดค่า compile program
- 4.5 การกำหนด ค่า run program

#### 5.1 การนำเข้าสู่บทเรียน

- 1.ครูเริ่มต้นด้วยการอธิบายถึงความสำคัญของการพัฒนาโปรแกรมจาวา และแนะนำเครื่องมือที่จำเป็นอย่าง JDK และโปรแกรม EditPlus
- 2.ครูแสดงวิดีโอสั้นที่อธิบายถึงความสามารถของ JDK และ EditPlus รวมถึงการใช้งานร่วมกันในการพัฒนาโปรแกรม
- 3.ถามตอบเพื่อประเมินความเข้าใจพื้นฐานของผู้เรียนเกี่ยวกับโปรแกรมที่จำเป็นต่อการพัฒนาโปรแกรมจาวา

#### 5.2 การเรียนรู้

##### กิจกรรมที่ 1: การติดตั้ง JDK (1 ชั่วโมง)

1. ครูสาธิตการดาวน์โหลดและติดตั้ง JDK จากเว็บไซต์ Oracle (หรือ AdoptOpenJDK)
2. อธิบายวิธีการตั้งค่า Environment Variables บนระบบปฏิบัติการ (Windows/Mac/Linux)
3. ผู้เรียนทำตามขั้นตอนที่ครูสอน โดยครูคอยดูแลและแก้ไขปัญหาเฉพาะหน้า
4. ผู้เรียนตรวจสอบการติดตั้ง JDK โดยใช้คำสั่ง `java -version` และ `javac -version` ผ่าน Command Line หรือ Terminal

##### กิจกรรมที่ 2: การติดตั้ง EditPlus (1 ชั่วโมง)

1. ครูสาธิตการดาวน์โหลดและติดตั้ง EditPlus
2. สอนวิธีการตั้งค่า EditPlus เพื่อรองรับการเขียนและคอมไพล์โค้ดภาษา Java
3. ผู้เรียนทำตามขั้นตอนการติดตั้งและตั้งค่า EditPlus โดยครูคอยตรวจสอบและช่วยแก้ปัญหา

##### กิจกรรมที่ 3: การทดสอบเขียนโค้ดด้วย EditPlus (2 ชั่วโมง)

1. ครูสาธิตการเขียนโปรแกรมจาวาอย่างง่ายใน EditPlus เช่น การพิมพ์ "Hello World"
2. ผู้เรียนเขียนโค้ดโปรแกรมจาวาและรันโค้ดโดยใช้ EditPlus
3. ครูช่วยแก้ไขปัญหาหากผู้เรียนพบข้อผิดพลาดในการคอมไพล์หรือรันโปรแกรม
4. ผู้เรียนฝึกเขียนโปรแกรมเพิ่มเติมโดยใช้ EditPlus เช่น การคำนวณทางคณิตศาสตร์อย่างง่าย

### 5.3 การสรุป

1. ครูสรุปขั้นตอนการติดตั้ง JDK และ EditPlus รวมถึงวิธีการตั้งค่าเพื่อให้ใช้งานร่วมกันได้
2. สอบถามผู้เรียนถึงข้อสังเกตและปัญหาที่พบในระหว่างการติดตั้งและทดสอบการเขียนโปรแกรม
3. ทบทวนการทำงานของ JDK และ EditPlus ว่ามีความสำคัญอย่างไรต่อการพัฒนาโปรแกรมจาวา

### 5.4 การวัดและประเมินผล

1. ประเมินผลจากใบงาน
2. แบบทดสอบ
3. สังเกต

## 6. สื่อการเรียนรู้/แหล่งการเรียนรู้

### 6.1 สื่อสิ่งพิมพ์

1. คอมพิวเตอร์พร้อมอินเทอร์เน็ต
2. ซอฟต์แวร์ JDK และ EditPlus

### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

### 6.3 หุ่นจำลองหรือของจริง (ถ้ามี)

.....

.....

### 6.4 อื่นๆ (ถ้ามี)

.....

.....

## 7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

## 8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....



## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 3

ชื่อหน่วย Classes และ Objects

สอนครั้งที่ 3

ชั่วโมงรวม 5

### 1.สาระสำคัญ

Class และ Object เป็นส่วนสำคัญของการเขียนโปรแกรมเชิงวัตถุ (OOP) โดย Class เป็นแม่แบบสำหรับสร้าง Object ที่แสดงถึงสิ่งที่จับต้องได้หรือแนวคิดที่มีความเป็นรูปธรรมในโลกแห่งความเป็นจริง Object เป็นตัวแทนของข้อมูลและการทำงานที่เกิดจาก class โดยแต่ละ object มีคุณลักษณะและพฤติกรรมที่แตกต่างกัน

### 2. สมรรถนะประจำหน่วย

- 2.1 เขียนโปรแกรมสร้างคลาสได้
- 2.2 เขียนโปรแกรมสร้าง ออบเจกต์ได้
- 2.3 เรียกใช้งาน ออบเจกต์ได้

### 3. จุดประสงค์การเรียนรู้

#### 3.1 ดานความรู้

- 3.1.1 เข้าใจแนวคิดพื้นฐานเกี่ยวกับ Class และ Object ในโปรแกรมเชิงวัตถุ (OOP)
- 3.1.2 อธิบายการทำงานของ constructor, method, และ field ภายใน class ได้

#### 3.2 ดานทักษะ

- 3.2.1 เขียนโปรแกรมเพื่อสร้างและใช้งาน Class และ Object ได้อย่างถูกต้อง
- 3.2.2 สร้างโปรแกรมจาวาที่ใช้ Object เพื่อจัดการข้อมูลและการทำงานของโปรแกรมได้

#### 3.3 คุณลักษณะที่พึงประสงค์

- 3.3.1 การตรงต่อเวลา
- 3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4. เนื้อหาสาระการเรียนรู้

- 4.1 การสร้างคลาส
- 4.2 การสร้าง ออบเจกต์
- 4.3 การสร้าง เมธอด
- 4.4 constructor

#### 5.1 การนำเขาสูบทเรียน

1 ครูเริ่มต้นด้วยการอธิบายถึงความสำคัญของโปรแกรมเชิงวัตถุ (OOP) ในการพัฒนาโปรแกรมขนาดใหญ่ โดยยกตัวอย่างระบบที่ใช้งานแนวคิด OOP เช่น ระบบธนาคาร ระบบการจัดการข้อมูลลูกค้า ฯลฯ

2 ครูอธิบายแนวคิดพื้นฐานของ Class และ Object โดยใช้ตัวอย่างจากชีวิตจริง เช่น การจำลองรถยนต์เป็น Object ซึ่งแต่ละคันเป็น Object ที่เกิดจาก Class ที่ชื่อว่า "รถยนต์"

3 ถามตอบเพื่อประเมินความเข้าใจเบื้องต้นของผู้เรียนเกี่ยวกับโปรแกรมเชิงวัตถุ

## 5.2 การเรียนรู้

### กิจกรรมที่ 1: การสร้าง Class และ Object เบื้องต้น (1 ชั่วโมง)

1. ครูอธิบายโครงสร้างของ Class และ Object ในภาษา Java โดยใช้ตัวอย่างโค้ดง่าย ๆ เช่น การสร้าง class ชื่อ Car ที่มี field เช่น color และ brand
2. ครูสอนวิธีการสร้าง Object จาก class โดยการใช้อคำสั่ง new เช่น Car myCar = new Car();
3. ผู้เรียนทดลองเขียนโค้ดเพื่อสร้าง Class และสร้าง Object ด้วยตนเอง โดยครูคอยดูแลและให้คำแนะนำ

### กิจกรรมที่ 2: การทำงานของ Constructor และ Method (1 ชั่วโมง)

1. ครูอธิบายการทำงานของ constructor ที่ใช้สำหรับการกำหนดค่าเริ่มต้นให้กับ Object เมื่อถูกสร้างขึ้น
2. สอนวิธีการสร้าง method ภายใน class และการเรียกใช้ method ผ่าน Object
3. ผู้เรียนทดลองสร้าง constructor และ method สำหรับ class ที่ตนเองสร้าง โดยมีการกำหนดค่าเริ่มต้นและการทำงานของ method

### กิจกรรมที่ 3: การฝึกปฏิบัติด้วยการใช้ Class และ Object (2 ชั่วโมง)

1. ครูมอบหมายให้ผู้เรียนออกแบบและเขียนโปรแกรมที่มี Class และ Object อย่างน้อย 2 class โดยอาจให้โจทย์ เช่น การสร้างระบบจัดการข้อมูลนักเรียนที่มี class นักเรียนและ class โรงเรียน
2. ผู้เรียนทำงานเป็นกลุ่มเพื่อออกแบบ class ที่สอดคล้องกับโจทย์ จากนั้นเขียนโปรแกรมและทดสอบการทำงานของ Object ที่สร้างขึ้น
3. ครูคอยให้คำแนะนำและตรวจสอบความถูกต้องของโค้ดที่ผู้เรียนเขียน

## 5.3 การสรุป

- 1 ครูสรุปเนื้อหาสำคัญเกี่ยวกับการสร้างและใช้งาน Class และ Object พร้อมยกตัวอย่างกรณีศึกษาเพิ่มเติม เพื่อให้ผู้เรียนเห็นภาพชัดเจน
- 2 สอบถามผู้เรียนเกี่ยวกับแนวคิดหลักที่ได้เรียนรู้และอธิบายถึงการนำไปใช้ในการเขียนโปรแกรมจริง
- 3 ผู้เรียนแชร์ประสบการณ์การทำงานเป็นกลุ่มและสรุปผลการเรียนรู้ของแต่ละกลุ่ม

## 5.4 การวัดและประเมินผล

- 1 การประเมินจากการสังเกต: สังเกตผู้เรียนในระหว่างการปฏิบัติงาน โดยตรวจสอบการสร้าง class, constructor, method และการใช้งาน Object ว่าทำงานถูกต้องหรือไม่
- 2 การประเมินจากผลงาน: ประเมินโค้ดที่ผู้เรียนเขียนว่าเป็นไปตามข้อกำหนดและสามารถทำงานได้ถูกต้องตามโจทย์ที่ได้รับ
- 3 การประเมินจากการถามตอบ: ประเมินความเข้าใจของผู้เรียนผ่านการถามตอบเกี่ยวกับแนวคิดของ Class และ Object เช่น ความแตกต่างระหว่าง Class กับ Object, หน้าที่ของ constructor และการทำงานของ method

4 แบบทดสอบหลังเรียน: ผู้เรียนทำแบบทดสอบข้อเขียนหรือออนไลน์เกี่ยวกับเนื้อหาของ Class และ Object เพื่อทบทวนและยืนยันความเข้าใจ

## 6.สื่อการเรียนรู้/แหล่งการเรียนรู้

### 6.1 สื่อสิ่งพิมพ์

- 1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต
- 2 ซอฟต์แวร์ JDK และ EditPlus

### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

### 6.3 ทุนจำลองหรือของจริง (ถ้ามี)

.....

.....

### 6.4 อื่นๆ (ถ้ามี)

.....

.....

## 7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

## 8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....



## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 4

ชื่อหน่วย Constructors และ Modifiers

สอนครั้งที่ 4

ชั่วโมงรวม 5

### 1.สาระสำคัญ

Constructors เป็น method พิเศษใน class ที่ถูกเรียกใช้เมื่อมีการสร้าง object ขึ้นมาเพื่อกำหนดค่าเริ่มต้นของ object ส่วน Modifiers เป็นคำสั่งที่ใช้กำหนดสิทธิ์ในการเข้าถึงและการปรับปรุงคุณสมบัติของ class, method และตัวแปร Access Modifiers เช่น public, private, protected กำหนดขอบเขตการเข้าถึง ในขณะที่ Non-Access Modifiers เช่น final, static, และ abstract กำหนดลักษณะการทำงานขององค์ประกอบต่างๆ

### 2.สมรรถนะประจำหน่วย

- 2.1 เข้าใจหลักการ Constructors และ Modifiers
- 2.2 เขียนโปรแกรมสร้าง Constructors และ Modifiers
- 2.3 การทดสอบแก้ไขและเรียกใช้

### 3. จุดประสงค์การเรียนรู้

#### 3.1 ดานความรู้

- 3.1.1 เข้าใจและอธิบายการทำงานของ Constructors ในภาษา Java

#### 3.2 ดานทักษะ

- 3.2.1 เขียนโปรแกรมที่ใช้ Constructors เพื่อกำหนดค่าเริ่มต้นของ Object ได้
- 3.2.2 นำแนวคิดของ Constructors และ Modifiers มาประยุกต์ใช้ในการออกแบบโปรแกรมเชิงวัตถุ

#### 3.3 คุณลักษณะที่พึงประสงค์

- 3.3.1 การตรงต่อเวลา
- 3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4. เนื้อหาสาระการเรียนรู้

#### 5.1 การนำเข้าสู่บทเรียน

- 1 ครูเริ่มต้นด้วยการอธิบายแนวคิดหลักเกี่ยวกับ Constructors และ Modifiers ในภาษา Java โดยใช้ตัวอย่างโค้ดที่มีการสร้าง object และกำหนดค่าเริ่มต้นผ่าน constructor
- 2 ยกตัวอย่างจากชีวิตจริง เช่น การจำลองการสร้างบัญชีผู้ใช้ใหม่ (object) ที่ต้องการค่าตั้งต้น เช่น ชื่อผู้ใช้และอีเมลผ่าน constructor
- 3 ครูอธิบายถึงบทบาทของ Modifiers โดยยกตัวอย่าง Access Modifiers (public, private, protected) และ Non-Access Modifiers (final, static, abstract) ว่ามีผลอย่างไรต่อความปลอดภัยและการทำงานของโปรแกรม

## 5.2 การเรียนรู้

### □ กิจกรรมที่ 1: การสร้างและการใช้ Constructors (1 ชั่วโมง)

1. ครูอธิบายถึงความสำคัญของ constructor ในการกำหนดค่าเริ่มต้นให้กับ object โดยอธิบายถึงการสร้าง default constructor และ parameterized constructor
2. ครูสาธิตการเขียนโค้ด Java เพื่อสร้าง constructor และการเรียกใช้ใน class ต่าง ๆ เช่น class User ที่มี constructor สำหรับตั้งค่า username และ email
3. ผู้เรียนเขียนโค้ดเพื่อสร้าง class และใช้ constructor ในการกำหนดค่าเริ่มต้น โดยครูคอยดูแลและให้คำแนะนำ

### 1 กิจกรรมที่ 2: การใช้งาน Access Modifiers (1 ชั่วโมง)

1. ครูสอนเกี่ยวกับ Access Modifiers (public, private, protected) ว่ามีผลต่อขอบเขตการเข้าถึงข้อมูลและ method อย่างไรใน class ต่าง ๆ
2. สาธิตการเขียนโค้ดที่แสดงการเข้าถึงตัวแปรและ method ใน class ผ่านการกำหนด Modifiers ที่แตกต่างกัน เช่น การใช้ private ในการซ่อนข้อมูลและการใช้ public ในการเปิดเผย method
3. ผู้เรียนฝึกการเขียนโค้ดโดยใช้ Access Modifiers เพื่อป้องกันข้อมูลและจัดการการเข้าถึงตัวแปรต่าง ๆ

### 2 กิจกรรมที่ 3: การใช้งาน Non-Access Modifiers (1 ชั่วโมง)

1. ครูอธิบายการทำงานของ Non-Access Modifiers เช่น final, static, และ abstract โดยยกตัวอย่างการใช้ static สำหรับ method หรือตัวแปรที่เป็นของ class และ final สำหรับป้องกันการเปลี่ยนแปลงค่าของตัวแปร
2. สาธิตการเขียนโค้ดที่แสดงการใช้ Non-Access Modifiers ในสถานการณ์ต่าง ๆ เช่น การใช้ final เพื่อสร้างค่าคงที่และ static เพื่อให้ method เป็นของ class โดยตรง
3. ผู้เรียนทดลองเขียนโปรแกรมโดยใช้ Non-Access Modifiers เพื่อควบคุมลักษณะการทำงานของตัวแปรและ method

### 3 กิจกรรมที่ 4: การฝึกปฏิบัติรวม (1 ชั่วโมง)

1. ครูให้โจทย์เพื่อทดสอบความเข้าใจของผู้เรียน โดยกำหนดให้ผู้เรียนออกแบบและเขียนโปรแกรมที่มีการใช้ Constructors และ Modifiers อย่างเหมาะสม
2. ผู้เรียนทำงานเป็นกลุ่มหรือรายบุคคลเพื่อพัฒนาโปรแกรมที่มีการใช้ constructor ในการกำหนดค่าเริ่มต้นของ object และใช้ Modifiers เพื่อควบคุมการเข้าถึงข้อมูล
3. ครูคอยให้คำแนะนำและตรวจสอบความถูกต้องของโค้ด

## 5.3 การสรุป

1. ครูสรุปเนื้อหาสำคัญเกี่ยวกับการทำงานของ Constructors และ Modifiers รวมถึงข้อดีของการใช้ Modifiers เพื่อควบคุมการเข้าถึงข้อมูลและป้องกันข้อผิดพลาดในโปรแกรม
2. ถามตอบและให้ผู้เรียนแชร์ประสบการณ์จากการฝึกปฏิบัติ เช่น ปัญหาที่พบและการแก้ไขปัญหา
3. ครูย้ำถึงความสำคัญของการออกแบบโปรแกรมเชิงวัตถุที่ดี โดยการใช้ Constructors และ Modifiers เพื่อสร้างโปรแกรมที่มีความยืดหยุ่นและความปลอดภัยในการทำงาน



## 5.4 การวัดและประเมินผล

1 การประเมินจากการสังเกต: สังเกตการทำงานของนักเรียนในระหว่างการฝึกปฏิบัติ โดยตรวจสอบว่านักเรียนสามารถสร้างและใช้ Constructors และ Modifiers ได้อย่างถูกต้องหรือไม่

2 การประเมินจากผลงาน: ประเมินโค้ดที่นักเรียนเขียน โดยพิจารณาการใช้งาน Constructors และ Modifiers ที่ถูกต้องและเหมาะสมตามข้อกำหนดของโจทย์

3 การประเมินจากการถามตอบ: สอบถามนักเรียนเกี่ยวกับบทบาทของ Constructors และ Modifiers ว่า4 แบบทดสอบหลังเรียน: นักเรียนทำแบบทดสอบข้อเขียนหรือออนไลน์เกี่ยวกับ Constructors และ Modifiers เพื่อวัดความเข้าใจในบทเรียน

## 6.สื่อการเรียนรู้/แหล่งการเรียนรู้

### 6.1 สื่อสิ่งพิมพ์

- 1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต
- 2 ซอฟต์แวร์ JDK และ EditPlus

### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

### 6.3 หุ่นจำลองหรือของจริง (ถ้ามี)

.....

.....

### 6.4 อื่นๆ (ถ้ามี)

.....


.....

## 7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

## 8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	
	วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006	หน่วยที่ 5
	ชื่อหน่วย Encapsulation และ Packages	สอนครั้งที่ 5-6
		ชั่วโมงรวม 10
<p><b>1.สาระสำคัญ</b></p> <p>Encapsulation เป็นแนวคิดในโปรแกรมเชิงวัตถุที่ใช้เพื่อปกป้องข้อมูลภายใน object โดยซ่อนข้อมูลบางส่วนไว้และเปิดเผยเฉพาะข้อมูลที่จำเป็นผ่าน method เท่านั้น โดยใช้ Access Modifiers เช่น private และ public เพื่อควบคุมการเข้าถึงข้อมูล</p> <p>Packages เป็นกลไกในการจัดกลุ่ม class และ interfaces ที่เกี่ยวข้องกันเพื่อให้โปรแกรมมีโครงสร้างชัดเจนและลดความซับซ้อนในการจัดการโค้ด</p> <p><b>2.สมรรถนะประจำหน่วย</b></p> <p>2.1 เข้าใจหลักการ Encapsulation และ Packages</p> <p>2.2 เขียนโปรแกรมสร้าง Encapsulation และ Packages</p> <p>2.3 การทดสอบแก้ไขและเรียกใช้</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p>3.1 ดานความรู้</p> <p>3.1.1 อธิบายแนวคิดและความสำคัญของ Encapsulation ในการพัฒนาโปรแกรมเชิงวัตถุ</p> <p>3.1.2 เข้าใจวิธีการใช้งาน Access Modifiers เพื่อสร้างการปกป้องข้อมูล (Data Protection) ในโปรแกรม</p> <p>3.2 ดานทักษะ</p> <p>3.2.1 ออกแบบและใช้งาน Packages เพื่อจัดการโครงสร้างโปรแกรมใน Java ได้อย่างมีประสิทธิภาพ</p> <p>3.2.2 พัฒนาความคิดสร้างสรรค์ในการออกแบบโปรแกรมที่ใช้ Encapsulation และ Packages ผ่านกระบวนการ Design Thinking</p> <p>3.3 คุณลักษณะที่พึงประสงค์</p> <p>3.3.1 การตรงต่อเวลา</p> <p>3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.</p> <p><b>4. เนื้อหาสาระการเรียนรู้</b></p> <p>4.1 หลักการ Encapsulation</p> <p>4.2 การใช้งาน Encapsulation</p>		

### 4.3 หลักการ Packages

### 4.4 การใช้งาน Packages

## 5.1 การนำเขาสอบทเรียน

1 ครูอธิบายภาพรวมของแนวคิด Encapsulation และ Packages ว่าเป็นแนวทางในการจัดการโปรแกรมเชิงวัตถุ โดยยกตัวอย่างจากการพัฒนาโปรแกรมในชีวิตจริง

2 อธิบายถึงปัญหาที่เกิดขึ้นหากไม่มีการปกป้องข้อมูลในโปรแกรม เช่น การที่ข้อมูลถูกแก้ไขโดยไม่ตั้งใจ

3 ครูยกตัวอย่างการใช้งาน Encapsulation ผ่านการใช้ Access Modifiers (private, public, protected) และอธิบายถึงโครงสร้างของ Packages ว่าเป็นเครื่องมือในการจัดการโค้ดในโปรแกรมขนาดใหญ่

## 5.2 การเรียนรู้

ใช้กระบวนการ **Design Thinking** เพื่อให้ผู้เรียนสามารถนำแนวคิดของ Encapsulation และ Packages ไปประยุกต์ใช้ในการพัฒนาโปรแกรมที่มีประสิทธิภาพ

### 1. Empathize (1 ชั่วโมง)

1. ครูให้ผู้เรียนทำการสำรวจและทำความเข้าใจเกี่ยวกับปัญหาของการไม่ใช้ Encapsulation เช่น การที่ข้อมูลในโปรแกรมถูกแก้ไขโดยไม่ได้รับอนุญาต
2. ผู้เรียนศึกษาตัวอย่างโค้ดที่ไม่มีการใช้ Encapsulation และวิเคราะห์ปัญหาที่อาจเกิดขึ้นจากการเข้าถึงข้อมูลอย่างไม่มีการควบคุม

### 2 Define (1 ชั่วโมง)

1. ครูให้ผู้เรียนร่วมกันกำหนดปัญหาที่พบจากการไม่ใช้ Encapsulation โดยกำหนดปัญหาหลักและผลกระทบที่เกิดขึ้น
2. ผู้เรียนวิเคราะห์ปัญหาและนำเสนอแนวทางการแก้ไขปัญหาด้วยการใช้ Encapsulation และการจัดการโครงสร้างโค้ดด้วย Packages

### 3 Ideate (2 ชั่วโมง)

1. ผู้เรียนระดมสมองในการออกแบบโปรแกรมที่ใช้แนวคิด Encapsulation เพื่อปกป้องข้อมูลและ Packages เพื่อจัดกลุ่ม class
2. กลุ่มนักเรียนแต่ละกลุ่มนำเสนอแนวทางการแก้ไขปัญหาด้วยการใช้ Encapsulation และ Packages พร้อมแผนการจัดการข้อมูลในโปรแกรม

### 4 Prototype (2 ชั่วโมง)

1. ผู้เรียนสร้างต้นแบบโปรแกรมโดยใช้ Encapsulation และ Packages ตามแนวทางที่ได้ออกแบบ
2. นักเรียนเขียนโปรแกรมที่ใช้ Access Modifiers และจัดโครงสร้างโค้ดด้วยการใช้ Packages เพื่อแยก class ต่างๆ ออกจากกันให้ชัดเจน
3. ครูคอยให้คำแนะนำในการพัฒนาโค้ดและตรวจสอบความถูกต้องของการใช้ Encapsulation และ Packages ในโปรแกรม

### 5 Test (2 ชั่วโมง)

1. ผู้เรียนทดสอบโปรแกรมที่สร้างขึ้นเพื่อดูว่าการใช้ Encapsulation สามารถปกป้องข้อมูลได้อย่างถูกต้องหรือไม่
2. กลุ่มนักเรียนทำการตรวจสอบและทดสอบการจัดการโค้ดผ่าน Packages ว่าสามารถจัดการได้อย่างมีประสิทธิภาพหรือไม่
3. ครูให้ข้อเสนอแนะเพิ่มเติมเกี่ยวกับการออกแบบและการปรับปรุงโค้ดตามผลการทดสอบ

### 5.3 การสรุป

1 ครูสรุปแนวคิดเกี่ยวกับ Encapsulation และ Packages โดยย้ำถึงข้อดีของการใช้ Encapsulation เพื่อปกป้องข้อมูลในโปรแกรม และการจัดการโครงสร้างโค้ดด้วย Packages เพื่อให้โปรแกรมมีความเป็นระเบียบและง่ายต่อการบำรุงรักษา

2 นักเรียนแบ่งปันประสบการณ์การทำงานเป็นกลุ่มและการใช้กระบวนการ Design Thinking ในการพัฒนาโปรแกรม

3 ครูเน้นย้ำถึงความสำคัญของการออกแบบโปรแกรมที่ยั่งยืนและความสามารถในการปรับปรุงได้ง่าย

### 5.4 การวัดและประเมินผล

1 การประเมินจากการสังเกต: สังเกตพฤติกรรมการทำงานของผู้เรียนในระหว่างกระบวนการ Design Thinking โดยเฉพาะการทำงานร่วมกันเป็นกลุ่ม

2 การประเมินจากผลงาน: ประเมินโค้ดโปรแกรมที่ผู้เรียนพัฒนา โดยพิจารณาการใช้ Encapsulation และ Packages ที่ถูกต้องและเหมาะสมตามหลักการออกแบบโปรแกรม

3 การประเมินจากการถามตอบ: สอบถามผู้เรียนเกี่ยวกับการใช้ Encapsulation และ Packages ว่ามีผลอย่างไรต่อการทำงานของโปรแกรม

4 แบบทดสอบหลังเรียน: ให้ผู้เรียนทำแบบทดสอบข้อเขียนหรือออนไลน์เกี่ยวกับแนวคิดของ Encapsulation และ Packages

## 6. สื่อการเรียนรู้/แหล่งการเรียนรู้

### 6.1 สื่อสิ่งพิมพ์

- 1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต
- 2 ซอฟต์แวร์ JDK และ EditPlus

### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

### 6.3 หุ่นจำลองหรือของจริง (ถ้ามี)

.....

.....

6.4 อื่นๆ (ถ้ามี)

.....  
.....

7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....



## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 6

ชื่อหน่วย Inheritance และ Polymorphism

สอนครั้งที่ 7-8

ชั่วโมงรวม 10

### 1.สาระสำคัญ

Inheritance เป็นแนวคิดในการสร้างคลาสใหม่จากคลาสที่มีอยู่แล้ว (superclass) เพื่อใช้คุณสมบัติที่มีอยู่ร่วมกันและขยายเพิ่มเติมตามความต้องการ

Polymorphism เป็นแนวคิดที่อนุญาตให้ method ต่าง ๆ ที่มีชื่อเดียวกันสามารถทำงานได้ในหลายรูปแบบตามคลาสที่สืบทอดมา หรือผ่านการใช้งาน Interface

Method Overriding ช่วยให้คลาสที่สืบทอดสามารถแก้ไขพฤติกรรมของ method ที่ได้จาก superclass เพื่อปรับให้เหมาะสมกับการทำงานใหม่

### 2.สมรรถนะประจำหน่วย

- 2.1 เข้าใจหลักการ Inheritance และ Polymorphism
- 2.2 เขียนโปรแกรมสร้าง Inheritance และ Polymorphism
- 2.3 การทดสอบแก้ไขและเรียกใช้

### 3. จุดประสงค์การเรียนรู้

#### 3.1 ดานความรู้

3.1.1 เข้าใจแนวคิดของ Inheritance (การสืบทอดคลาส) และสามารถนำไปประยุกต์ใช้ในการเขียนโปรแกรมเชิงวัตถุได้

3.1.2 อธิบายถึงแนวคิดของ Polymorphism (พหุสัณฐาน) ใน Java และเข้าใจการใช้งาน Method Overriding และ Interface

#### 3.2 ดานทักษะ

3.2.1 เขียนโปรแกรมที่ใช้ Inheritance และ Polymorphism เพื่อสร้างโครงสร้างโปรแกรมที่ยืดหยุ่นและง่ายต่อการขยาย

#### 3.3 คุณลักษณะที่พึงประสงค์

3.3.1 การตรงต่อเวลา

3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4. เนื้อหาสาระการเรียนรู้

- 4.1 แนวคิดและหลักการ Inheritance
- 4.2 superclass
- 4.3 subclass

#### 4.4 แนวคิดและหลักการ Polymorphism

#### 4.5 Method Overriding

#### 4.6. Method Overloading

### 5.1 การนำเขาสูบทเรียน

1 ครูเริ่มต้นด้วยการอธิบายแนวคิดพื้นฐานเกี่ยวกับ Inheritance และ Polymorphism ผ่านตัวอย่างชีวิตจริง เช่น การสืบทอดลักษณะจากบรรพบุรุษในสิ่งมีชีวิต (Inheritance) และการทำงานในหลากหลายรูปแบบของเครื่องมือเดียวกัน (Polymorphism)

2 แสดงตัวอย่างโปรแกรมใน Java ที่ใช้ Inheritance ในการสร้างคลาสใหม่จากคลาสเดิม พร้อมอธิบายโครงสร้างโค้ดและความหมายของคำสั่งต่าง ๆ

3 ครูอธิบายหลักการของ Polymorphism โดยเน้นการใช้งานผ่าน Method Overriding และการประยุกต์ใช้ Interface เพื่อสร้างความยืดหยุ่นในโปรแกรม

### 5.2 การเรียนรู้

ใช้กระบวนการ แบ่งกลุ่มศึกษาและนำเสนอหน้าชั้นเรียน เพื่อให้ผู้เรียนมีโอกาสฝึกการทำงานร่วมกันและการนำเสนอความรู้ที่ได้รับ

#### ขั้นที่ 1: แบ่งกลุ่มศึกษา (2 ชั่วโมง)

1. แบ่งผู้เรียนออกเป็นกลุ่ม (กลุ่มละ 3-5 คน) แต่ละกลุ่มรับผิดชอบหัวข้อที่แตกต่างกันเกี่ยวกับ

Inheritance และ Polymorphism เช่น:

- กลุ่มที่ 1: การสร้างคลาสและการสืบทอดคลาสใน Inheritance
- กลุ่มที่ 2: การใช้งาน Method Overriding ใน Polymorphism
- กลุ่มที่ 3: การประยุกต์ใช้ Interface เพื่อสร้าง Polymorphism

2. ผู้เรียนในแต่ละกลุ่มศึกษาหัวข้อที่ได้รับมอบหมาย โดยครูเป็นที่ปรึกษาและให้คำแนะนำในการเข้าใจหัวข้อ รวมถึงการใช้งานโค้ดตัวอย่าง

#### ขั้นที่ 2: พัฒนาเนื้อหาและฝึกนำเสนอ (2 ชั่วโมง)

1. ผู้เรียนในแต่ละกลุ่มพัฒนาเนื้อหาและโค้ดตัวอย่างเกี่ยวกับหัวข้อที่ศึกษา พร้อมทำสไลด์นำเสนอเพื่ออธิบายความรู้ที่ได้เรียนรู้

2. แต่ละกลุ่มฝึกการนำเสนอเนื้อหาที่ตนเองศึกษาให้เพื่อนร่วมกลุ่มฟังเพื่อเตรียมพร้อมสำหรับการนำเสนอหน้าชั้นเรียน

#### ขั้นที่ 3: นำเสนอหน้าชั้นเรียน (3 ชั่วโมง)

1. แต่ละกลุ่มนำเสนอผลงานหน้าชั้นเรียน โดยมีเวลาในการนำเสนอและสาธิตการทำงานของโปรแกรมที่ใช้ Inheritance และ Polymorphism

2. ครูและเพื่อนร่วมชั้นเรียนร่วมกันให้คำแนะนำและเสนอแนวทางการปรับปรุง โดยเน้นความชัดเจนและความเข้าใจของผู้เรียนในแนวคิดที่นำเสนอ

3. ครูสรุปและเสริมความเข้าใจให้แก่ผู้เรียนหลังการนำเสนอของแต่ละกลุ่ม โดยยกตัวอย่างเพิ่มเติมหากจำเป็น

#### ชั้นที่ 4: การฝึกปฏิบัติร่วมกัน (1 ชั่วโมง)

1. ผู้เรียนทุกคนทำงานร่วมกันในโปรแกรมที่มีการใช้งาน Inheritance และ Polymorphism โดยครูให้โจทย์โปรแกรมที่ครอบคลุมแนวคิดทั้งสอง เช่น การออกแบบคลาสพนักงานที่มีการสืบทอดจากคลาสบุคลากร พร้อมการใช้ Polymorphism เพื่อจัดการหน้าที่ของพนักงานหลายรูปแบบ
2. ครูคอยให้คำแนะนำและตรวจสอบการทำงานของนักเรียนในแต่ละขั้นตอน

#### 5.3 ชั้นสรุป

- 1 ครูสรุปเนื้อหาหลักที่ได้เรียนรู้เกี่ยวกับ Inheritance และ Polymorphism โดยเน้นถึงความสำคัญของการใช้ทั้งสองแนวคิดในการออกแบบโปรแกรมที่สามารถขยายและปรับปรุงได้ง่าย
- 2 ครูให้ผู้เรียนทบทวนตัวอย่างโปรแกรมที่ได้พัฒนา พร้อมให้แต่ละกลุ่มสะท้อนความคิดเห็นเกี่ยวกับสิ่งที่ได้เรียนรู้จากการทำงานเป็นกลุ่มและการนำเสนอ
- 3 ครูย้ำถึงแนวคิดของการใช้ Inheritance เพื่อการขยายความสามารถของโปรแกรม และการใช้ Polymorphism เพื่อเพิ่มความยืดหยุ่นในการทำงานของ method ต่าง ๆ

#### 5.4 การวัดและประเมินผล

- 1 การประเมินจากการสังเกต: สังเกตการทำงานร่วมกันของผู้เรียนในกลุ่ม การแบ่งหน้าที่และการช่วยเหลือกันในระหว่างการศึกษาและนำเสนอ
- 2 การประเมินจากผลงาน: ประเมินคุณภาพของเนื้อหาและโค้ดที่แต่ละกลุ่มนำเสนอ โดยพิจารณาความถูกต้องและความสมบูรณ์ของการใช้งาน Inheritance และ Polymorphism ในโปรแกรม
- 3 การประเมินจากการนำเสนอ: ประเมินความชัดเจนและความสามารถในการอธิบายแนวคิดที่ผู้เรียนได้นำเสนอหน้าชั้นเรียน
- 4 แบบทดสอบหลังเรียน: ผู้เรียนทำแบบทดสอบข้อเขียนหรือออนไลน์เพื่อวัดความเข้าใจในหัวข้อ Inheritance และ Polymorphism

#### 6. สื่อการเรียนรู้/แหล่งการเรียนรู้

##### 6.1 สื่อสิ่งพิมพ์

- 1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต
- 2 ซอฟต์แวร์ JDK และ EditPlus

##### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

##### 6.3 ทุนจำลองหรือของจริง (ถ้ามี)

.....

.....



6.4 อื่นๆ (ถ้ามี)

.....  
.....

7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....



## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 7

ชื่อหน่วย Abstraction และ Interface

สอนครั้งที่ 9-10

ชั่วโมงรวม 10

### 1.สาระสำคัญ

Abstraction เป็นกระบวนการซ่อนรายละเอียดการทำงานที่ซับซ้อนและแสดงเฉพาะส่วนที่จำเป็นเท่านั้นในโปรแกรม โดยสามารถทำได้ผ่าน Abstract Classes และ Methods ที่ไม่มีการกำหนดรายละเอียดการทำงานไว้ล่วงหน้า

Interface เป็นโครงสร้างที่กำหนดการทำงานที่จำเป็นโดยไม่มีการกำหนดวิธีการทำงาน ผู้พัฒนาโปรแกรมสามารถนำไปใช้ได้ตามความเหมาะสมเพื่อให้โปรแกรมมีความยืดหยุ่นมากขึ้น

### 2.สมรรถนะประจำหน่วย

- 2.1 เข้าใจหลักการ Abstraction และ Interface
- 2.2 เขียนโปรแกรมสร้าง Abstraction และ Interface
- 2.3 -การทดสอบแก้ไขและเรียกใช้

### 3. จุดประสงค์การเรียนรู้

#### 3.1 ดานความรู้

3.1.1 เข้าใจแนวคิดของ Abstraction (การแยกความซับซ้อน) และ Interface ในการพัฒนาโปรแกรมเชิงวัตถุ

#### 3.2 ดานทักษะ

3.2.1 ใช้ Abstract Classes และ Interfaces เพื่อสร้างโครงสร้างโปรแกรมที่ยืดหยุ่นและง่ายต่อการขยาย

3.2.2 เขียนโปรแกรมที่ใช้หลักการ Abstraction และ Interface เพื่อลดความซับซ้อนของโค้ดและเพิ่มความสามารถในการขยายโปรแกรม

3.2.3 วิเคราะห์และออกแบบโปรแกรมที่ใช้ Abstraction และ Interface ผ่านการทำงานเป็นกลุ่ม และนำเสนอผลงานต่อเพื่อนร่วมชั้น

#### 3.3 คุณลักษณะที่พึงประสงค์

3.3.1 การตรงต่อเวลา

3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4. เนื้อหาสาระการเรียนรู้

4.1 หลักการและแนวคิด Abstraction

4.2 Abstract Classes

### 4.3 Abstract Methods

### 4.4 การประยุกต์ใช้ Interface

## 5.1 การนำเขาสอบทเรียน

1 ครูอธิบายแนวคิดของ Abstraction และ Interface ในการเขียนโปรแกรม โดยยกตัวอย่างสถานการณ์จริงที่มีการใช้การแยกส่วนความซับซ้อนเพื่อให้ผู้เรียนเข้าใจง่ายขึ้น

2 แสดงตัวอย่างโปรแกรมที่ใช้ Abstract Classes และ Interface ใน Java พร้อมกับอธิบายการใช้งานของแต่ละส่วน

3 สอบถามความคิดเห็นและให้ผู้เรียนตั้งคำถามเกี่ยวกับการใช้ Abstraction และ Interface เพื่อกระตุ้นการมีส่วนร่วม

## 5.2 การเรียนรู้

การเรียนรู้เป็นรายบุคคล เน้นการฝึกปฏิบัติเป็นรายบุคคล โดยมีครูเป็นที่ปรึกษาและให้คำแนะนำ

### ขั้นที่ 1: การศึกษาและทำความเข้าใจ (2 ชั่วโมง)

1. ครูแจกเอกสารหรือแหล่งข้อมูลที่เกี่ยวข้องกับ Abstraction และ Interface ให้ผู้เรียนศึกษา
2. ครูอธิบายโค้ดตัวอย่างเพิ่มเติมในส่วนของ Abstraction และ Interface ใน Java เพื่อให้ผู้เรียนมีความเข้าใจที่ชัดเจนขึ้น
3. ผู้เรียนทำแบบฝึกหัดที่เกี่ยวข้องกับการสร้าง Abstract Class และ Interface ในโปรแกรมเล็กๆ ที่กำหนด

### ขั้นที่ 2: การฝึกปฏิบัติ (3 ชั่วโมง)

1. ผู้เรียนเริ่มเขียนโปรแกรมโดยนำแนวคิด Abstraction และ Interface มาใช้ในการออกแบบคลาสในโปรแกรมของตนเอง เช่น การสร้างระบบจัดการยานพาหนะที่มีคลาสรถยนต์ จักรยาน เป็นต้น
2. ครูให้โจทย์โปรแกรมเพิ่มเติมที่ผู้เรียนต้องนำแนวคิดทั้งสองมาใช้เพื่อแก้ไขปัญหาคือการออกแบบโปรแกรม
3. ผู้เรียนพัฒนาโปรแกรมตามโจทย์ที่ได้รับ โดยมีครูคอยตรวจสอบและให้คำแนะนำเป็นระยะ ๆ

### ขั้นที่ 3: การทบทวนและปรับปรุง (2 ชั่วโมง)

1. ผู้เรียนทบทวนโปรแกรมที่พัฒนาขึ้นเอง พร้อมปรับปรุงโค้ดให้มีความชัดเจนและทำงานได้อย่างถูกต้องตามแนวคิด Abstraction และ Interface
2. ครูชี้แนะจุดบกพร่องของโค้ดและแนะนำวิธีปรับปรุงให้มีประสิทธิภาพมากขึ้น พร้อมอธิบายเพิ่มเติมเกี่ยวกับการใช้ Abstraction และ Interface อย่างมีประสิทธิภาพในโปรแกรมขนาดใหญ่

### ขั้นที่ 4: การนำไปประยุกต์ใช้ (1 ชั่วโมง)

1. ครูมอบหมายให้ผู้เรียนออกแบบโครงสร้างโปรแกรมที่ซับซ้อนขึ้น ซึ่งต้องใช้ทั้ง Abstract Class และ Interface เพื่อจัดการหลายคลาสและการทำงานที่เกี่ยวข้อง
2. ผู้เรียนประยุกต์ใช้แนวคิดที่ได้เรียนรู้เพื่อลดความซับซ้อนของโค้ดและเพิ่มความยืดหยุ่นของโปรแกรม

## 5.3 ชั้นสรุป

1 ครูสรุปหลักการสำคัญของ Abstraction และ Interface โดยเน้นความแตกต่างและการประยุกต์ใช้ในสถานการณ์ต่าง ๆ

2 ครูตอบคำถามและเปิดโอกาสให้ผู้เรียนแบ่งปันประสบการณ์การเขียนโปรแกรมตามแนวคิดที่ได้อ่าน

3 ครูย้ำถึงความสำคัญของการใช้ Abstraction เพื่อซ่อนรายละเอียดที่ซับซ้อน และการใช้ Interface เพื่อกำหนดพฤติกรรมที่สอดคล้องกันในหลาย ๆ คลาส

#### 5.4 การวัดและประเมินผล

1 การประเมินจากการสังเกต: ครูสังเกตความสามารถในการเขียนโค้ดและการใช้แนวคิด Abstraction และ Interface ในโปรแกรมของผู้เรียน

2 การประเมินจากผลงาน: ประเมินจากโค้ดที่ผู้เรียนพัฒนาขึ้น โดยพิจารณาความถูกต้องและประสิทธิภาพของการใช้ Abstract Class และ Interface

3 การทดสอบภาคปฏิบัติ: ผู้เรียนทำข้อสอบเขียนโปรแกรมที่เกี่ยวข้องกับการใช้ Abstraction และ Interface ในการแก้ปัญหาโปรแกรม

4 แบบฝึกหัดหลังเรียน: ผู้เรียนทำแบบฝึกหัดเพิ่มเติมหรือแบบทดสอบออนไลน์ เพื่อวัดความเข้าใจแนวคิด Abstraction และ Interface

### 6. สื่อการเรียนรู้/แหล่งการเรียนรู้

#### 6.1 สื่อสิ่งพิมพ์

- 1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต
- 2 ซอฟต์แวร์ JDK และ EditPlus

#### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

#### 6.3 หุ่นจำลองหรือของจริง (ถ้ามี)

.....

.....

#### 6.4 อื่นๆ (ถ้ามี)

.....

.....

### 7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....



## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 8

ชื่อหน่วย Exception Handling และ Threads

สอนครั้งที่ 11-12

ชั่วโมงรวม 10

### 1.สาระสำคัญ

Exception Handling คือกระบวนการจัดการกับข้อผิดพลาดที่เกิดขึ้นขณะทำงานของโปรแกรม โดยไม่ทำให้โปรแกรมหยุดทำงานทันที ซึ่งสามารถใช้ try-catch-finally ใน Java เพื่อจัดการกับข้อผิดพลาดที่เกิดขึ้น

Threads คือส่วนหนึ่งของโปรแกรมที่สามารถทำงานพร้อมกันได้หลายส่วน (Multithreading) ซึ่งช่วยเพิ่มประสิทธิภาพการทำงานของโปรแกรม โดยสามารถใช้คลาส Thread หรือ Runnable ในการจัดการ Threads

### 2.สมรรถนะประจำหน่วย

- 2.1 เข้าใจหลักการภาษา Exception Handling
- 2.2 เข้าใจหลักการ Threads
- 2.3 เขียนโปรแกรมสร้าง Exception Handling และ Threads
- 2.4 การทดสอบแก้ไขและเรียกใช้

### 3. จุดประสงค์การเรียนรู้

#### 3.1 ดานความรู้

- 3.1.1 อธิบายแนวคิดของ Exception Handling และ Threads ใน Java
- 3.1.2 เข้าใจการทำงานแบบ Multithreading และสามารถสร้างและจัดการ Threads ใน Java ได้

#### 3.2 ดานทักษะ

- 3.2.1 เขียนโปรแกรมที่จัดการกับข้อผิดพลาด (Exceptions) อย่างถูกต้องและมีประสิทธิภาพ
- 3.2.2 ประยุกต์ใช้แนวคิด Exception Handling และ Threads ในการเขียนโปรแกรมที่ซับซ้อน

#### 3.3 คุณลักษณะที่พึงประสงค์

- 3.3.1 การตรงต่อเวลา
- 3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4.เนื้อหาสาระการเรียนรู้

- 4.1 หลักการและแนวคิด Exception Handling
- 4.2 คำสั่ง try catch
- 4.3 คำสั่ง throw throws และ finally
- 4.4 หลักการทำงานของ Threads

#### 5.1 การนำเขาสูบทเรียน

1 ครูเริ่มต้นอธิบายแนวคิดพื้นฐานของ **Exception Handling** โดยยกตัวอย่างข้อผิดพลาดที่พบบ่อยในโปรแกรม เช่น การแบ่งด้วยศูนย์ หรือการเข้าถึงไฟล์ที่ไม่พบ

2 ครูแสดงตัวอย่างโค้ด Java ที่มีการจัดการข้อยกเว้นโดยใช้ try-catch-finally และอธิบายการทำงานของโค้ด

3 ครูนำเข้าสู่หัวข้อ **Threads** โดยอธิบายความหมายของการทำงานพร้อมกัน (Multithreading) และยกตัวอย่างสถานการณ์จริง เช่น การดาวน์โหลดไฟล์ขณะที่ผู้ใช้ยังสามารถทำงานอื่นได้ในโปรแกรม

## 5.2 การเรียนรู้

### กระบวนการเรียนรู้ด้วยตนเองพร้อมการสนับสนุนจากครู

ผู้เรียนจะฝึกปฏิบัติเป็นรายบุคคล โดยมีครูคอยสนับสนุนและให้คำแนะนำในทุกขั้นตอน

#### • **ขั้นที่ 1: การศึกษาและทำความเข้าใจ (2 ชั่วโมง)**

1. ครูแจกเอกสารหรือแหล่งข้อมูลเกี่ยวกับ Exception Handling และ Threads ให้ผู้เรียนศึกษา
2. ครูสาธิตเพิ่มเติมเกี่ยวกับวิธีการจัดการข้อยกเว้นในโปรแกรม โดยใช้ตัวอย่างที่ซับซ้อนขึ้น เช่น การจัดการกับ Exceptions หลายระดับ (Multiple Exceptions)
3. ผู้เรียนเริ่มฝึกเขียนโค้ดเพื่อจัดการข้อยกเว้นในสถานการณ์ต่าง ๆ โดยใช้โค้ดตัวอย่างที่ได้รับจากครู

#### **ขั้นที่ 2: การฝึกปฏิบัติการสร้างและจัดการ Threads (3 ชั่วโมง)**

1. ครูอธิบายโครงสร้างของการสร้าง Threads ใน Java ผ่านการใช้คลาส Thread และ Runnable
2. ผู้เรียนฝึกเขียนโปรแกรมที่ใช้ Multithreading ในการจำลองการทำงานพร้อมกัน เช่น การคำนวณค่าต่าง ๆ และการพิมพ์ข้อมูลพร้อมกัน
3. ผู้เรียนทดลองสร้าง Threads และจัดการกับการทำงานของ Threads เช่น การหยุด, การระงับ, และการต่อเนื้องานของ Threads ผ่านคำสั่ง sleep(), join(), interrupt()

#### **ขั้นที่ 3: การนำ Exception Handling และ Threads มาประยุกต์ใช้ร่วมกัน (2 ชั่วโมง)**

1. ผู้เรียนฝึกเขียนโปรแกรมที่ซับซ้อนขึ้น โดยต้องจัดการข้อยกเว้น (Exceptions) ร่วมกับการจัดการ Threads เช่น การสร้างโปรแกรมที่คำนวณตัวเลขหลายส่วนพร้อมกันและจัดการกับข้อยกเว้นที่เกิดขึ้นในแต่ละ Thread
2. ครูคอยให้คำแนะนำเมื่อผู้เรียนพบปัญหาหรือข้อสงสัยในขณะที่เขียนโปรแกรม

#### **ขั้นที่ 4: การแก้ไขและปรับปรุงโปรแกรม (1 ชั่วโมง)**

1. ผู้เรียนทบทวนและปรับปรุงโปรแกรมที่พัฒนาขึ้นตามคำแนะนำของครู
2. ครูตรวจสอบโค้ดของผู้เรียนและชี้แนะวิธีการปรับปรุงให้มีประสิทธิภาพมากขึ้น ทั้งในส่วนของจัดการข้อยกเว้นและการทำงานของ Threads

## 5.3 ขั้นสรุป

1 ครูสรุปแนวคิดหลักของ Exception Handling และ Threads โดยเน้นประโยชน์ในการพัฒนาโปรแกรมที่มีความเสถียรและมีประสิทธิภาพ

2 ครูตอบคำถามที่เกี่ยวข้องและย้ำถึงความสำคัญของการใช้ Exception Handling เพื่อลดการเกิดปัญหาในโปรแกรม และการใช้ Threads เพื่อเพิ่มประสิทธิภาพของการทำงานที่ต้องทำงานพร้อมกัน

3 ผู้เรียนทำแบบฝึกหัดหรืออภิปรายเกี่ยวกับแนวทางการปรับปรุงโปรแกรมที่ได้เขียน

#### 5.4 การวัดและประเมินผล

1 การประเมินจากการสังเกต: ครูสังเกตความสามารถของผู้เรียนในการเขียนโค้ดที่ใช้ Exception Handling และ Threads อย่างถูกต้องและมีประสิทธิภาพ

2 การประเมินจากผลงาน: ประเมินจากโค้ดที่ผู้เรียนพัฒนาขึ้น โดยพิจารณาความถูกต้องในการจัดการข้อผิดพลาดและการทำงานของ Threads

3 การทดสอบภาคปฏิบัติ: ผู้เรียนทำข้อสอบเขียนโปรแกรมที่เกี่ยวข้องกับการใช้ Exception Handling และ Threads ในสถานการณ์ต่าง ๆ

4 แบบฝึกหัดหลังเรียน: ผู้เรียนทำแบบฝึกหัดเพิ่มเติมหรือแบบทดสอบออนไลน์ เพื่อวัดความเข้าใจแนวคิด Exception Handling และ Threads

#### 6. สื่อการเรียนรู้/แหล่งการเรียนรู้

##### 6.1 สื่อสิ่งพิมพ์

1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต

2 ซอฟต์แวร์ JDK และ EditPlus

##### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

##### 6.3 หุ่นจำลองหรือของจริง (ถ้ามี)

.....

.....

##### 6.4 อื่นๆ (ถ้ามี)

.....

.....

#### 7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

#### 8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....





## แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

วิชา การพัฒนาซอฟต์แวร์เชิงวัตถุ รหัสวิชา 31901-2006

หน่วยที่ 9

ชื่อหน่วย การเชื่อมต่อนฐานข้อมูลและการประยุกต์ใช้งานอาชีพ

สอนครั้งที่ 13-14

ชั่วโมงรวม 10

### 1.สาระสำคัญ

**การเชื่อมต่อนฐานข้อมูล** เป็นส่วนสำคัญของการพัฒนาโปรแกรมในระดับอาชีพ โดยในภาษา Java สามารถใช้ JDBC เพื่อเชื่อมต่อกับฐานข้อมูลประเภทต่าง ๆ เช่น MySQL, PostgreSQL หรือ SQLite เพื่อจัดการข้อมูลในโปรแกรมได้

**การประยุกต์ใช้ในงานอาชีพ** หมายถึงการนำความรู้เรื่องการเชื่อมต่อนฐานข้อมูลไปประยุกต์ใช้ในการพัฒนาระบบที่ตอบสนองความต้องการในงานจริง เช่น การพัฒนาแอปพลิเคชันสำหรับการจัดการข้อมูลในองค์กร

### 2.สมรรถนะประจำหน่วย

- 2.1 เข้าใจหลักการของการเชื่อมต่อนฐานข้อมูล
- 2.2 เขียนโปรแกรมเชื่อมต่อนฐานข้อมูลได้
- 2.3 เขียนโปรแกรมจัดการข้อมูลในฐานข้อมูลได้

### . จุดประสงค์การเรียนรู้

#### 3.1 ดานความรู้

3.1.1 มีความรู้เข้าใจหลักการเชื่อมต่อนฐานข้อมูลกับโปรแกรม Java และวิธีการใช้ JDBC (Java Database Connectivity)

#### 3.2 ดานทักษะ

- 3.2.1เขียนโปรแกรมที่สามารถเชื่อมต่อนฐานข้อมูลได้อย่างมีประสิทธิภาพ
- 3.2.2 จัดการกับข้อมูลในฐานข้อมูล เช่น การเพิ่ม, ลบ, แก้ไข และเรียกดูข้อมูลผ่านโปรแกรม
- 3.2.3 ประยุกต์ใช้การเชื่อมต่อนฐานข้อมูลในการพัฒนาโปรแกรมที่สอดคล้องกับความต้องการในงานอาชีพ เช่น ระบบจัดการข้อมูลลูกค้า หรือระบบจัดการสินค้าคงคลัง

#### 3.3 คุณลักษณะที่พึงประสงค์

- 3.3.1 การตรงต่อเวลา
- 3.3.2 ปฏิบัติงานด้วยความประณีต รอบคอบ ปลอดภัย ตระหนักถึงคุณภาพของงาน.

### 4.เนื้อหาสาระการเรียนรู้

- 4.1 การเขียนโปรแกรมเชื่อมต่อนฐานข้อมูล
- 4.2 การเขียนโปรแกรมเพิ่มข้อมูล
- 4.3 การเขียนโปรแกรมแก้ไขข้อมูล
- 4.4 การเขียนโปรแกรมลบข้อมูล
- 4.5 การเขียนโปรแกรมแสดงผลข้อมูล

## 5.1 การนำเขาสู่บทเรียน

1 ครูเริ่มต้นอธิบายแนวคิดพื้นฐานของ การเชื่อมต่อฐานข้อมูล โดยให้ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล และการจัดเก็บข้อมูลในรูปแบบต่าง ๆ

2 ครูแนะนำเกี่ยวกับ JDBC ซึ่งเป็น API สำหรับเชื่อมต่อฐานข้อมูลในภาษา Java พร้อมยกตัวอย่างการใช้งานในงานอาชีพจริง เช่น ระบบการจัดการร้านค้า

3 ครูสาธิตตัวอย่างโค้ดที่ใช้ JDBC ในการเชื่อมต่อกับฐานข้อมูล MySQL หรือ SQLite และอธิบาย ส่วนประกอบของโค้ด

## 5.2 การเรียนรู้

### กระบวนการเรียนรู้แบบขั้นตอนการพัฒนาและการนำไปประยุกต์ใช้งาน

ผู้เรียนจะฝึกปฏิบัติผ่านขั้นตอนการพัฒนาโปรแกรมที่เชื่อมต่อกับฐานข้อมูล โดยครูคอยให้คำแนะนำตลอดกระบวนการ

#### ขั้นที่ 1: การศึกษาและทำความเข้าใจ (2 ชั่วโมง)

1. ครูแจกเอกสารเกี่ยวกับ JDBC และแหล่งข้อมูลเพิ่มเติมเกี่ยวกับการเชื่อมต่อฐานข้อมูลใน Java
2. ครูสาธิตเพิ่มเติมเกี่ยวกับการจัดการข้อมูลในฐานข้อมูล เช่น การเพิ่มข้อมูลใหม่ (INSERT), การแก้ไขข้อมูล (UPDATE), และการลบข้อมูล (DELETE)
3. ผู้เรียนเริ่มต้นการเขียนโปรแกรมที่เชื่อมต่อกับฐานข้อมูล พร้อมสร้างคำสั่ง SQL เพื่อจัดการกับข้อมูล

#### ขั้นที่ 2: การฝึกปฏิบัติการจัดการข้อมูลในฐานข้อมูล (3 ชั่วโมง)

1. ผู้เรียนฝึกปฏิบัติในการเขียนโปรแกรมที่สามารถดึงข้อมูลจากฐานข้อมูลและแสดงผลในโปรแกรม
2. ผู้เรียนฝึกเพิ่ม, แก้ไข และลบข้อมูลในฐานข้อมูลผ่านโปรแกรม โดยใช้คำสั่ง SQL ที่ส่งผ่าน JDBC
3. ครูให้คำแนะนำในการออกแบบโปรแกรมที่เชื่อมต่อฐานข้อมูลได้อย่างมีประสิทธิภาพและตอบสนองความต้องการของงานจริง เช่น การพัฒนาระบบการจัดการลูกค้า

#### ขั้นที่ 3: การประยุกต์ใช้ในงานอาชีพ (2 ชั่วโมง)

1. ผู้เรียนประยุกต์ใช้ความรู้ที่ได้เรียนรู้มาพัฒนาโปรแกรมที่มีการเชื่อมต่อฐานข้อมูลในบริบทของงานอาชีพ เช่น ระบบจัดการสินค้าคงคลัง, ระบบจัดการพนักงาน หรือระบบการจองสินค้าผ่านออนไลน์
2. ครูมอบหมายโครงการขนาดเล็กให้ผู้เรียนพัฒนาโปรแกรมที่มีการเชื่อมต่อฐานข้อมูล โดยใช้ทักษะที่ได้เรียนรู้มาในการออกแบบโครงสร้างและการจัดการฐานข้อมูล

#### ขั้นที่ 4: การแก้ไขและปรับปรุงโปรแกรม (1 ชั่วโมง)

1. ผู้เรียนทบทวนและปรับปรุงโปรแกรมที่พัฒนาขึ้นตามคำแนะนำของครู
2. ครูตรวจสอบโค้ดและให้คำแนะนำเกี่ยวกับการปรับปรุงประสิทธิภาพของโปรแกรม ทั้งในส่วนของการเชื่อมต่อฐานข้อมูลและการจัดการข้อมูล

## 5.3 ชั้นสรุป

1 ครูสรุปเนื้อหาเกี่ยวกับการเชื่อมต่อฐานข้อมูลและการจัดการข้อมูลผ่าน JDBC โดยเน้นความสำคัญในการพัฒนาโปรแกรมที่สามารถทำงานกับฐานข้อมูลได้

2 ครูตอบคำถามที่เกี่ยวข้องกับการเชื่อมต่อฐานข้อมูลและการใช้ JDBC ในการพัฒนาโปรแกรม พร้อมอธิบายตัวอย่างการประยุกต์ใช้ในการทำงานอาชีพ

3 ผู้เรียนทำแบบฝึกหัดเพื่อทบทวนความรู้และประเมินความเข้าใจ

#### 5.4 การวัดและประเมินผล

1 การประเมินจากการสังเกต: ครูสังเกตความสามารถของผู้เรียนในการเขียนโค้ดที่เชื่อมต่อด้านข้อมูลอย่างถูกต้อง

2 การประเมินจากผลงาน: ประเมินจากโค้ดที่ผู้เรียนพัฒนาขึ้น โดยพิจารณาความถูกต้องในการเชื่อมต่อด้านข้อมูลและการจัดการข้อมูลผ่านโปรแกรม

3 การทดสอบภาคปฏิบัติ: ผู้เรียนทำข้อสอบเขียนโปรแกรมที่เกี่ยวข้องกับการเชื่อมต่อด้านข้อมูลและการจัดการข้อมูลผ่าน JDBC

4 แบบฝึกหัดหลังเรียน: ผู้เรียนทำแบบฝึกหัดเพิ่มเติมหรือแบบทดสอบออนไลน์ เพื่อวัดความเข้าใจแนวคิดการเชื่อมต่อด้านข้อมูลและการประยุกต์ใช้ในงานอาชีพ

#### 6. สื่อการเรียนรู้/แหล่งการเรียนรู้

##### 6.1 สื่อสิ่งพิมพ์

- 1 คอมพิวเตอร์พร้อมอินเทอร์เน็ต
- 2 ซอฟต์แวร์ JDK และ EditPlus

##### 6.2 สื่อโสตทัศน (ถ้ามี)

.....

.....

##### 6.3 หุ่นจำลองหรือของจริง (ถ้ามี)

.....

.....

##### 6.4 อื่นๆ (ถ้ามี)

.....

.....

#### 7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ )

.....

#### 8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น

.....

