

แบบฝึกออกแบบโมดูลการเรียนรู้

หลักสูตรประกาศนียบัตรวิชาชีพ (ปวช.)

การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น

รหัสวิชา 21910-1003

ผู้จัดทำ

นายสรวิษฐ์ หนูมาศ

วิทยาลัยอาชีวศึกษานครศรีธรรมราช

สำนักงานคณะกรรมการการอาชีวศึกษา

กระทรวงศึกษาธิการ

หลักสูตรรายวิชา

หลักสูตรประกาศนียบัตรวิชาชีพ (ปวช.)

ประเภทวิชา อุตสาหกรรมดิจิทัลและเทคโนโลยีสารสนเทศ กลุ่มอาชีพ ธุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ สาขาวิชา
เทคโนโลยีธุรกิจดิจิทัล

รหัสวิชา 21910-1003 ชื่อวิชา การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 2 ชั่วโมง/สัปดาห์ รวม 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

อ้างอิงมาตรฐาน

- มาตรฐานคุณวุฒิวิชาชีพ สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) อาชีพนักพัฒนาระบบ ระดับ 3

ผลลัพธ์การเรียนรู้ระดับรายวิชา

เขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ได้อย่างเป็นระบบ ด้วยความละเอียด รอบคอบ

จุดประสงค์รายวิชา เพื่อให้

1. เข้าใจหลักการเขียนโปรแกรมคอมพิวเตอร์
2. มีทักษะในการวิเคราะห์ ออกแบบผังงาน รหัสเทียม และการพัฒนาโปรแกรม
3. มีความสามารถประยุกต์ใช้คำสั่งการทำงานคอมพิวเตอร์เบื้องต้น
4. มีเจตคติและกิจนิสัยที่ดีในการปฏิบัติงานด้วยความละเอียด รอบคอบ

สมรรถนะรายวิชา

1. แสดงความรู้เกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์ตามหลักการ
2. ออกแบบผังงาน เขียนรหัสเทียมตามหลักการ
3. เขียนคำสั่งการทำงานโปรแกรมด้วยภาษาคอมพิวเตอร์ตามหลักการและกระบวนการ
4. ประยุกต์ใช้คำสั่งการทำงานคอมพิวเตอร์เบื้องต้นในงานธุรกิจ

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการเขียนโปรแกรมคอมพิวเตอร์ การแก้ปัญหการทำงาน การวิเคราะห์ปัญหา การกำหนดขั้นตอนการทำงาน (Algorithm) ด้วยภาษาคอมพิวเตอร์ การเขียนผังงาน (Flowchart) การเขียนรหัสเทียม (Pseudocode) หลักการออกแบบโปรแกรม และปฏิบัติการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์

โครงสร้างโมดูลการเรียนรู้

รหัสวิชา 21910-1003 ชื่อวิชา การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น

เวลาเรียนรวม 54 ชั่วโมง (ทฤษฎี 18 ชั่วโมง ปฏิบัติ 36 ชั่วโมง)

ที่	ชื่อโมดูล	เวลาเรียน (ชม.)			หมายเหตุ
		ทฤษฎี	ปฏิบัติ	รวม	
1	หลักการเขียนโปรแกรมและการวิเคราะห์ปัญหา	2	4	6	
2	อัลกอริทึม ผังงาน และรหัสเทียม	3	6	9	
3	คำสั่งพื้นฐาน ตัวแปร ชนิดข้อมูล และการรับ-แสดงผล	3	6	9	
4	คำสั่งเลือกทำและเงื่อนไข	3	6	9	
5	คำสั่งวนซ้ำและการจัดโครงสร้างโปรแกรม	3	6	9	
6	การพัฒนาโปรแกรมจากผังงานและการทดสอบแบบ Unit Test	3	6	9	
	ประเมินผลสัมฤทธิ์การเรียนรู้ปลายรายวิชา	1	2	3	ภาคทฤษฎี/ภาคปฏิบัติ
	รวม	18	36	54	

ตารางเชื่อมโยงโมดูลกับมาตรฐานอาชีพ

โมดูล	มาตรฐาน/สมรรถนะย่อยที่เชื่อมโยง	หลักฐานสำคัญ	วิธีประเมินหลัก
1	10301.01 (อ่าน/ทบทวน Work Flow)	ใบวิเคราะห์ IPO และลำดับงาน	ข้อเขียน + สัมภาษณ์ + ตรวจสอบผลงาน
2	10301.01 (เขียน Flow Chart)	Flowchart + Pseudocode + Trace Table	ข้อเขียน + สัมภาษณ์ + สาคิต
3	10301.02 (เขียนโปรแกรมตาม Flowchart)	Source Code + Test Data	สาคิตการรัน + ตรวจสอบโค้ด
4	10301.02 (เขียนโปรแกรมตาม Flowchart)	โปรแกรมเงื่อนไข + Test Case	สาคิต + สัมภาษณ์
5	10301.02 (เขียนโปรแกรมตาม Flowchart)	โปรแกรม Loop/Function + Trace Table	สาคิต + ตรวจสอบโค้ด
6	10301.01 10301.02 10302.01 10302.02	Mini Project + Test Plan + Unit Test Report	ข้อเขียน + สาคิต + สัมภาษณ์

โมดูลการเรียนรู้ที่ 1 หลักการเขียนโปรแกรมและการวิเคราะห์ปัญหา

เวลาเรียน 6 ชั่วโมง (ทฤษฎี 2 ชั่วโมง ปฏิบัติ 4 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้ + ทักษะ + จิตพิสัย = การประยุกต์ใช้)

ผู้เรียนอธิบายหลักการพัฒนาโปรแกรม วิเคราะห์ปัญหาอย่างเป็นระบบ ระบุข้อมูลนำเข้า กระบวนการ และผลลัพธ์ (Input-Process-Output) พร้อมกำหนดลำดับขั้นตอนการแก้ปัญหาเบื้องต้นได้ถูกต้อง โดยปฏิบัติงานด้วยความละเอียด รอบคอบ และตรวจสอบความสมเหตุสมผลก่อนนำไปพัฒนาโปรแกรม

2. อ้างอิงมาตรฐาน

- UoC 10301 ดำเนินการพัฒนาโปรแกรม
- EoC 10301.01 แปลงหน้าจอ (GUI) หรือ ลำดับงาน (Work Flow) เป็นผังงาน (Flow Chart) - เชื่อมโยงการอ่านและทบทวนลำดับงานก่อนเขียนผังงาน

3. สมรรถนะประจำโมดูล

- 3.1 แสดงความรู้เกี่ยวกับองค์ประกอบและกระบวนการพัฒนาโปรแกรมคอมพิวเตอร์
- 3.2 วิเคราะห์โจทย์และจำแนก Input-Process-Output ได้ตามเงื่อนไข
- 3.3 จัดลำดับขั้นตอนการแก้ปัญหาและตรวจสอบความครบถ้วนของลำดับงาน
- 3.4 ประยุกต์การวิเคราะห์ปัญหากับโจทย์งานธุรกิจเบื้องต้น

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

- 4.1.1 อธิบายความหมายของโปรแกรม โปรแกรมเมอร์ อัลกอริทึม และกระบวนการแก้ปัญหาได้
- 4.1.2 อธิบายความสัมพันธ์ระหว่าง Input-Process-Output กับการออกแบบโปรแกรมได้
- 4.1.3 จำแนกข้อกำหนด ข้อมูลนำเข้า เงื่อนไข กระบวนการ และผลลัพธ์จากโจทย์ที่กำหนดได้

4.2 ทักษะพิสัย

- 4.2.1 จัดทำตารางวิเคราะห์ปัญหาแบบ Input-Process-Output จากสถานการณ์ที่กำหนดได้
- 4.2.2 เรียงลำดับขั้นตอนการทำงานจากโจทย์ธุรกิจอย่างมีเหตุผลและตรวจสอบย้อนกลับได้

4.3 จิตพิสัย

- 4.3.1 ปฏิบัติงานอย่างเป็นระบบ มีความละเอียด รอบคอบ และรับผิดชอบต่อความถูกต้องของข้อมูล
- 4.3.2 แลกเปลี่ยนความคิดเห็นและทบทวนลำดับงานร่วมกับผู้อื่นอย่างสร้างสรรค์

4.4 การประยุกต์ใช้

- 4.4.1 วิเคราะห์กระบวนการคำนวณยอดขายสุทธิ/ส่วนลด/ภาษีจากสถานการณ์จำลองและจัดทำข้อกำหนดสำหรับนำไปออกแบบโปรแกรม

5. สารการเรียนรู้

- 5.1 ความหมายและความสำคัญของการเขียนโปรแกรมคอมพิวเตอร์
- 5.2 ขั้นตอนการแก้ปัญหา ทำความเข้าใจปัญหา วิเคราะห์ กำหนดวิธีแก้ ตรวจสอบ และปรับปรุง
- 5.3 การวิเคราะห์ Input-Process-Output และข้อกำหนดของโปรแกรม
- 5.4 คุณลักษณะของขั้นตอนวิธีที่ดี ชัดเจน เป็นลำดับ สิ้นสุดได้ และตรวจสอบได้
- 5.5 แนวคิดความผิดพลาดเบื้องต้น Syntax Runtime และ Logic Error ในฐานะสิ่งที่ต้องป้องกันตั้งแต่การออกแบบ

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ภาคทฤษฎี เวลาเรียน 2 ชั่วโมง

- ศึกษาหลักการพัฒนาโปรแกรมและวงจรการแก้ปัญหาจากสื่อประกอบ
- วิเคราะห์ตัวอย่างโจทย์และอภิปราย Input-Process-Output
- ทำแบบฝึกตรวจสอบความเข้าใจและสรุปแนวทางการวิเคราะห์ปัญหาเป็นขั้นตอน

6.2 ภาคปฏิบัติ เวลาเรียน 4 ชั่วโมง

- สาธิตการแยกข้อกำหนดจากโจทย์ธุรกิจ
- ฝึกทำใบวิเคราะห์ปัญหาและตรวจทานกับเพื่อน (peer review)
- ปรับแก้ลำดับงานจากข้อเสนอแนะและส่งเป็นหลักฐานก่อนเข้าสู่โมดูลฝึกงาน

6.3 รูปแบบกิจกรรมการเรียนรู้

- ก่อนเรียน แบบทดสอบ/คำถามกระตุ้นความรู้เดิมและชี้แจงภาระงาน
- ระหว่างเรียน อธิบายสั้น-สาธิต-ฝึกตามตัวอย่าง-ฝึกแก้โจทย์-ตรวจทานและรับข้อเสนอแนะ
- หลังเรียน สรุปแนวคิด ส่งชิ้นงาน/หลักฐาน ทำแบบทดสอบหลังเรียน และสะท้อนข้อผิดพลาดที่พบ

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี)

- สไลด์/เอกสารเรื่องหลักการเขียนโปรแกรมและกระบวนการแก้ปัญหา
- ใบความรู้เรื่อง Input-Process-Output และการวิเคราะห์ข้อกำหนด
- ตัวอย่างสถานการณ์งานธุรกิจ เช่น การขายสินค้า การคิดส่วนลด การคำนวณค่าบริการ
- แบบฝึกหัดก่อนเรียนและหลังเรียนบน Platform ของรายวิชา

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 4 ชั่วโมง

ฝึกปฏิบัติจริง

- ใบปฏิบัติงาน วิเคราะห์โจทย์ “ระบบคำนวณยอดชำระสินค้าหลังส่วนลดและภาษี”

หลักฐานการปฏิบัติงาน

- ใบวิเคราะห์ Input-Process-Output
- รายการเงื่อนไข/ข้อกำหนดของโปรแกรม
- ลำดับขั้นตอนการแก้ปัญหามอบปรับปรุงหลังการตรวจทาน

เกณฑ์การปฏิบัติงาน

- ระบุ Input-Process-Output ครบถ้วนและตรงโจทย์ไม่น้อยกว่า 80%
- ลำดับงานไม่มีขั้นตอนสำคัญตกหล่นและไม่ขัดแย้งกับเงื่อนไขโจทย์
- เอกสารอ่านเข้าใจง่าย ใช้คำสม่าเสมอ และมีการตรวจทานก่อนส่ง

ขอบเขตการปฏิบัติงาน

- ปฏิบัติงานเดี่ยวจากโจทย์ธุรกิจ 1 สถานการณ์ ภายใน 120 นาที และทบทวนร่วมกับเพื่อนตามแบบตรวจรายการ

วิธีการประเมิน

- ตรวจสอบผลงานจากใบงาน
- ซักถาม/สัมภาษณ์เหตุผลในการกำหนดขั้นตอน
- สังเกตความเป็นระบบและความรอบคอบระหว่างปฏิบัติ

เครื่องมือประเมิน

- แบบประเมินใบวิเคราะห์ปัญหา
- แบบตรวจรายการ (Checklist) ความครบถ้วนของข้อกำหนด
- แบบสังเกตพฤติกรรมการทำงาน

8. การวัดและประเมินผลประจำโมดูล

กำหนดสัดส่วนตามโครงสร้างฝึกเดิม ภาคทฤษฎี 30 คะแนน และภาคปฏิบัติ 70 คะแนน รวม 100 คะแนน

องค์ประกอบการประเมิน	คะแนน	หลักฐาน/เครื่องมือ
แบบทดสอบก่อนเรียน-หลังเรียน	20	แบบทดสอบความรู้
แบบฝึกวิเคราะห์โจทย์ระหว่างเรียน	10	ใบงาน/การมีส่วนร่วม
ทักษะการวิเคราะห์ปัญหาและจัดลำดับงาน	40	แบบประเมินการทำงาน
ชิ้นงานวิเคราะห์ IPO และข้อกำหนด	30	รูบริกส์ชิ้นงาน
รวม	100	

เกณฑ์พิจารณาผลงาน

- ความถูกต้องตามหลักการ/ผังงานและเงื่อนไขที่กำหนด
- ความสามารถปฏิบัติจริงและอธิบายเหตุผลของวิธีทำ
- คุณภาพหลักฐาน เช่น Source Code Flowchart Test Case Test Report
- ความละเอียด รอบคอบ ความเป็นระเบียบ และการส่งงานตามกำหนด

โมดูลการเรียนรู้ที่ 2 อัลกอริทึม ผังงาน และรหัสเทียม

เวลาเรียน 9 ชั่วโมง (ทฤษฎี 3 ชั่วโมง ปฏิบัติ 6 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้ + ทักษะ + จิตพิสัย = การประยุกต์ใช้)

ผู้เรียนสามารถแปลงลำดับงานหรือข้อกำหนดของระบบเป็นอัลกอริทึม ผังงาน (Flowchart) และรหัสเทียม (Pseudocode) ที่สื่อความหมายตรงกัน อ่านเข้าใจได้ และตรวจสอบแก้ไขตามเงื่อนไขที่กำหนด โดยยึดหลักสัญลักษณ์และลำดับตรรกะอย่างถูกต้อง

2. อ้างอิงมาตรฐาน

- UoC 10301 ดำเนินการพัฒนาโปรแกรม
- EoC 10301.01 แปลงหน้าจอ (GUI) หรือ ลำดับงาน (Work Flow) เป็นผังงาน (Flow Chart)
- PC อ่านแบบร่าง/ลำดับงาน ทบทวนและแก้ไข และเขียนผังงานตามที่กำหนด

3. สมรรถนะประจำโมดูล

- 3.1 อธิบายหลักการของ Algorithm Flowchart และ Pseudocode
- 3.2 เลือกใช้สัญลักษณ์ผังงานได้ถูกต้องตามหน้าที่
- 3.3 เขียนผังงานและรหัสเทียมจาก Work Flow/โจทย์ที่กำหนด
- 3.4 ตรวจสอบความสอดคล้องระหว่างโจทย์ ลำดับงาน ผังงาน และรหัสเทียม

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

- 4.1.1 บอกหน้าที่ของสัญลักษณ์ Flowchart พื้นฐานได้
- 4.1.2 อธิบายโครงสร้างลำดับ (Sequence) การตัดสินใจ (Selection) และการทำซ้ำ (Iteration) ในระดับแนวคิดได้
- 4.1.3 อธิบายหลักการเขียน Pseudocode ที่อ่านง่ายและไม่ผูกกับภาษาคอมพิวเตอร์ใดภาษาหนึ่งได้

4.2 ทักษะพิสัย

- 4.2.1 เขียน Flowchart จาก Work Flow ที่กำหนดได้
- 4.2.2 เขียน Pseudocode ให้สอดคล้องกับ Flowchart ได้
- 4.2.3 ทบทวนและแก้ไข Flowchart เมื่อพบเงื่อนไขตกหล่นหรือเส้นทางการทำงานไม่ครบ

4.3 จิตพิสัย

- 4.3.1 ใช้สัญลักษณ์และคำอธิบายอย่างสม่ำเสมอ มีระเบียบ และตรวจทานผลงานก่อนส่ง
- 4.3.2 ยอมรับข้อเสนอแนะและปรับแก้แบบร่างอย่างมีเหตุผล

4.4 การประยุกต์ใช้

- 4.4.1 ออกแบบ Flowchart และ Pseudocode สำหรับกระบวนการสมัครสมาชิก/คำนวณสิทธิส่วนลดหรือกระบวนการธุรกิจที่มีเงื่อนไข

5. สารการเรียนรู้

- 5.1 หลักการและรูปแบบการเขียน Algorithm
- 5.2 สัญลักษณ์ Flowchart Start/End Process Input/Output Decision Connector และทิศทางการไหล
- 5.3 โครงสร้างพื้นฐานของผังงาน Sequence Selection Iteration
- 5.4 หลักการเขียน Pseudocode และการกำหนดชื่อข้อมูล
- 5.5 การตรวจสอบผังงานด้วยการไล่ค่าทดสอบ (Desk Check/Trace)
- 5.6 การแปลง Work Flow หรือแบบร่างขั้นตอนการทำงานให้เป็น Flowchart

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ภาคทฤษฎี เวลาเรียน 3 ชั่วโมง

- เรียนรู้สัญลักษณ์และกติกาการเขียน Flowchart
- วิเคราะห์ตัวอย่าง Flowchart ที่ถูกต้อง/ผิดและอธิบายสาเหตุ
- ฝึกอ่าน Flowchart และเขียน Pseudocode จากตัวอย่าง

6.2 ภาคปฏิบัติ เวลาเรียน 6 ชั่วโมง

- รับ Work Flow จากผู้สอนแล้วระบุจุดตัดสินใจและข้อมูลที่เกี่ยวข้อง
- เขียน Flowchart ฉบับร่าง ตรวจสอบด้วย Trace Table และแก้ไข
- เขียน Pseudocode ให้ตรงกับ Flowchart และนำเสนอเหตุผลของการออกแบบ

6.3 รูปแบบกิจกรรมการเรียนรู้

- ก่อนเรียน แบบทดสอบ/คำถามกระตุ้นความรู้เดิมและชี้แจงภาระงาน
- ระหว่างเรียน อธิบายสั้น-สาธิต-ฝึกตามตัวอย่าง-ฝึกแก้โจทย์-ตรวจทานและรับข้อเสนอแนะ
- หลังเรียน สรุปแนวคิด ส่งชิ้นงาน/หลักฐาน ทำแบบทดสอบหลังเรียน และสะท้อนข้อผิดพลาดที่พบ

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี)

- สื่อสรุปสัญลักษณ์ Flowchart
- ใบความรู้ Algorithm และ Pseudocode
- ซอฟต์แวร์วาดผังงานหรือเครื่องมือ Diagram ที่สถานศึกษากำหนด
- ตัวอย่าง Work Flow/GUI จำลองสำหรับฝึกแปลงเป็นผังงาน

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 6 ชั่วโมง

ฝึกปฏิบัติจริง

- ใบปฏิบัติงาน แปลง Work Flow “ระบบรับคำสั่งซื้อและคำนวณส่วนลดสมาชิก” เป็น Flowchart และ Pseudocode

หลักฐานการปฏิบัติงาน

- ไฟล์/ภาพ Flowchart ฉบับสมบูรณ์
- เอกสาร Pseudocode
- Trace Table แสดงการทดสอบอย่างน้อย 3 กรณี

เกณฑ์การปฏิบัติงาน

- ใช้สัญลักษณ์ Flowchart ถูกต้องและเส้นทางการทำงานครบทุกเงื่อนไข
- Pseudocode สอดคล้องกับ Flowchart และอ่านเข้าใจง่าย
- Trace Table ครอบคลุมกรณีปกติและกรณีขอบเขตที่กำหนด

ขอบเขตการปฏิบัติงาน

- ปฏิบัติงานเดี่ยวหรือคู่จาก Work Flow 1 เรื่อง ภายใน 180 นาที พร้อมนำเสนอและตอบคำถามเกี่ยวกับผังงาน

วิธีการประเมิน

- ข้อสอบข้อเขียนเรื่องสัญลักษณ์/ตรรกะ
- สัมภาษณ์การอ่านผังงาน
- สาธิตการปรับแก้ Flowchart จากเงื่อนไขเพิ่มเติม

เครื่องมือประเมิน

- แบบทดสอบความรู้
- รูปrikส์ Flowchart และ Pseudocode
- แบบประเมินการสาธิต/สัมภาษณ์

8. การวัดและประเมินผลประจำโมดูล

กำหนดสัดส่วนตามโครงสร้างฝึกเดิม ภาคทฤษฎี 30 คะแนน และภาคปฏิบัติ 70 คะแนน รวม 100 คะแนน

องค์ประกอบการประเมิน	คะแนน	หลักฐาน/เครื่องมือ
แบบทดสอบความรู้ Flowchart/Pseudocode	20	แบบทดสอบ
แบบฝึกอ่านและตรวจผังงาน	10	ใบงาน
ทักษะการแปลง Work Flow เป็น Flowchart	40	แบบประเมินปฏิบัติ
ชิ้นงาน Flowchart + Pseudocode + Trace Table	30	รูปrikส์ชิ้นงาน
รวม	100	

เกณฑ์พิจารณาผลงาน

- ความถูกต้องตามหลักการ/ผังงานและเงื่อนไขที่กำหนด
- ความสามารถปฏิบัติจริงและอธิบายเหตุผลของวิธีทำ
- คุณภาพหลักฐาน เช่น Source Code Flowchart Test Case Test Report
- ความละเอียด รอบคอบ ความเป็นระเบียบ และการส่งงานตามกำหนด

โมดูลการเรียนรู้ที่ 3 คำสั่งพื้นฐาน ตัวแปร ชนิดข้อมูล และการรับ-แสดงผล

เวลาเรียน 9 ชั่วโมง (ทฤษฎี 3 ชั่วโมง ปฏิบัติ 6 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้ + ทักษะ + จิตพิสัย = การประยุกต์ใช้)

ผู้เรียนเขียนโปรแกรมพื้นฐานจากผังงานที่กำหนด โดยใช้ตัวแปร ชนิดข้อมูล ตัวดำเนินการ การรับข้อมูล และการแสดงผลได้ถูกต้องตามหลักภาษาและกระบวนการ พร้อมตรวจสอบผลลัพธ์กับข้อมูลทดสอบอย่างเป็นระบบ

2. อ่างอิงมาตรฐาน

- UoC 10301 ดำเนินการพัฒนาโปรแกรม
- EoC 10301.02 พัฒนาโปรแกรมตามผังงาน (Flow Chart)
- PC อ่านผังงานและลำดับงาน และเขียนโปรแกรมตามผังงาน

3. สมรรถนะประจำโมดูล

- 3.1 อธิบายโครงสร้างโปรแกรมและองค์ประกอบของคำสั่งพื้นฐาน
- 3.2 ประกาศ/กำหนดค่าและเลือกชนิดข้อมูลให้เหมาะสม
- 3.3 ใช้ตัวดำเนินการคำนวณและจัดลำดับนิพจน์ได้
- 3.4 รับข้อมูล ประมวลผล และแสดงผลตามผังงานที่กำหนด
- 3.5 ทดสอบโปรแกรมด้วยข้อมูลตัวอย่างและแก้ไขข้อผิดพลาดพื้นฐาน

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

- 4.1.1 อธิบายความหมายของตัวแปร ค่าคงที่ ชนิดข้อมูล นิพจน์ และตัวดำเนินการได้
- 4.1.2 อธิบายความแตกต่างระหว่างข้อมูลข้อความ จำนวนเต็ม จำนวนจริง และตรรกะในระดับพื้นฐานได้
- 4.1.3 คาดการณ์ผลลัพธ์ของนิพจน์และลำดับการคำนวณได้

4.2 ทักษะพิสัย

- 4.2.1 เขียนโปรแกรมรับข้อมูลและแสดงผลได้ตามรูปแบบ
- 4.2.2 เขียนนิพจน์คำนวณจากสูตรธุรกิจและจัดเก็บผลในตัวแปรได้
- 4.2.3 อ่าน Flowchart แบบลำดับและแปลงเป็นโค้ดที่ทำงานได้

4.3 จิตพิสัย

- 4.3.1 ตั้งชื่อตัวแปรสื่อความหมาย จัดรูปแบบโค้ดอ่านง่าย และตรวจสอบค่าที่ใช้คำนวณ
- 4.3.2 บันทึกเวอร์ชันงานและแก้ไขข้อผิดพลาดอย่างมีขั้นตอน

4.4 การประยุกต์ใช้

- 4.4.1 สร้างโปรแกรมคำนวณยอดสุทธิจากราคาสินค้า จำนวน ส่วนลด และภาษี พร้อมแสดงผลในรูปแบบที่อ่านง่าย

5. สารการเรียนรู้

- 5.1 โครงสร้างโปรแกรมและการทำงานแบบลำดับ (Sequence)
- 5.2 ตัวแปร ค่าคงที่ และหลักการจัดตั้งชื่อ
- 5.3 ชนิดข้อมูลพื้นฐานและการแปลงชนิดข้อมูลเบื้องต้น
- 5.4 ตัวดำเนินการทางคณิตศาสตร์ การกำหนดค่า การเปรียบเทียบ และตรรกะในภาพรวม
- 5.5 การรับข้อมูล (Input) และการแสดงผล (Output)

5.6 การเขียนนิพจน์และการจัดลำดับการคำนวณ

5.7 การอ่าน Error message และการแก้ Syntax/Runtime Error เบื้องต้น

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ภาคทฤษฎี เวลาเรียน 3 ชั่วโมง

- ทบทวน Flowchart แบบ Sequence แล้วเชื่อมกับคำสั่งโปรแกรม
- เรียนรู้ตัวแปร ชนิดข้อมูล ตัวดำเนินการ และการรับ/แสดงผลจากตัวอย่าง
- ทำแบบฝึกหัดการณั้ผลลัพธ์และค้นหาข้อผิดพลาดจากโค้ดสั้น ๆ

6.2 ภาคปฏิบัติ เวลาเรียน 6 ชั่วโมง

- สาธิตการสร้างโปรแกรมตาม Flowchart ทีละขั้น
- ฝึกเขียนโปรแกรมคำนวณหลายสูตรและตรวจผลด้วย Test Data
- ปรับปรุงชื่อ ตัวแปร รูปแบบการแสดงผล และความอ่านง่ายของโค้ด

6.3 รูปแบบกิจกรรมการเรียนรู้

- ก่อนเรียน แบบทดสอบ/คำถามกระตุ้นความรู้เดิมและชี้แจงภาระงาน
- ระหว่างเรียน อธิบายสั้น-สาธิต-ฝึกตามตัวอย่าง-ฝึกแก้โจทย์-ตรวจทานและรับข้อเสนอแนะ
- หลังเรียน สรุปแนวคิด ส่งชิ้นงาน/หลักฐาน ทำแบบทดสอบหลังเรียน และสะท้อนข้อผิดพลาดที่พบ

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี)

- IDE/Editor และภาษาคอมพิวเตอร์ที่สถานศึกษากำหนด (สามารถใช้ Python สำหรับการฝึกพื้นฐาน)
- ใบความรู้ตัวแปร ชนิดข้อมูล ตัวดำเนินการ และ I/O
- ตัวอย่าง Flowchart แบบ Sequence
- แบบฝึกแก้ข้อผิดพลาดและชุดข้อมูลทดสอบ

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 6 ชั่วโมง

ฝึกปฏิบัติจริง

- ใบปฏิบัติงาน พัฒนาโปรแกรม “คำนวณยอดสุทธิใบสั่งซื้อ” จาก Flowchart

หลักฐานการปฏิบัติงาน

- ไฟล์ Source Code
- Flowchart ต้นทางที่ใช้พัฒนา
- ตารางข้อมูลทดสอบและผลลัพธ์อย่างน้อย 5 กรณี

เกณฑ์การปฏิบัติงาน

- โปรแกรมทำงานตาม Flowchart และคำนวณถูกต้องตามสูตร
- ใช้ชนิดข้อมูลและตัวแปรเหมาะสม ชื่อตัวแปรสื่อความหมาย
- ผลลัพธ์จัดรูปแบบอ่านง่าย และผ่านข้อมูลทดสอบที่กำหนด

ขอบเขตการปฏิบัติงาน

- พัฒนาโปรแกรม Console/พื้นฐาน 1 โปรแกรมจาก Flowchart ภายใน 180 นาที โดยใช้โครงสร้างลำดับเป็นหลัก

วิธีการประเมิน

- ตรวจสอบการทำงานจริงของโปรแกรม

- สาธิตการรันด้วยข้อมูลทดสอบ
- ชักถามความหมายของตัวแปรและนิพจน์สำคัญ

เครื่องมือประเมิน

- แบบประเมิน Source Code
- Checklist ผลการทดสอบ
- แบบสัมภาษณ์สั้นระหว่างสาธิต

8. การวัดและประเมินผลประจำโมดูล

กำหนดสัดส่วนตามโครงสร้างฝึกเดิม ภาคทฤษฎี 30 คะแนน และภาคปฏิบัติ 70 คะแนน รวม 100 คะแนน

องค์ประกอบการประเมิน	คะแนน	หลักฐาน/เครื่องมือ
แบบทดสอบตัวแปร/ชนิดข้อมูล/ตัวดำเนินการ	20	แบบทดสอบ
แบบฝึกหัดการณั้ผลลัพธ์และแก้ Error	10	ใบงาน
ทักษะการเขียนโปรแกรมตาม Flowchart	40	แบบประเมินปฏิบัติ
ชิ้นงานโปรแกรมคำนวณยอดสุทธิ	30	รูปกริกส์ Source Code/ผลลัพธ์
รวม	100	

เกณฑ์พิจารณาผลงาน

- ความถูกต้องตามหลักการ/ผังงานและเงื่อนไขที่กำหนด
- ความสามารถปฏิบัติจริงและอธิบายเหตุผลของวิธีทำ
- คุณภาพหลักฐาน เช่น Source Code Flowchart Test Case Test Report
- ความละเอียด รอบคอบ ความเป็นระเบียบ และการส่งงานตามกำหนด

โมดูลการเรียนรู้ที่ 4 คำสั่งเลือกทำและเงื่อนไข

เวลาเรียน 9 ชั่วโมง (ทฤษฎี 3 ชั่วโมง ปฏิบัติ 6 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้ + ทักษะ + จิตพิสัย = การประยุกต์ใช้)

ผู้เรียนเขียนโปรแกรมที่มีการตัดสินใจจากเงื่อนไขแบบทางเลือกเดียว สองทาง และหลายทางตามผังงานได้อย่างถูกต้อง เลือกตัวดำเนินการเปรียบเทียบ/ตรรกะเหมาะสม และทดสอบทุกเส้นทางสำคัญของโปรแกรม

2. อ้างอิงมาตรฐาน

- UoC 10301 ดำเนินการพัฒนาโปรแกรม
- EoC 10301.02 พัฒนาโปรแกรมตามผังงาน (Flow Chart)
- PC อ่านผังงานและลำดับงาน และเขียนโปรแกรมตามผังงาน

3. สมรรถนะประจำโมดูล

- 3.1 อธิบายหลักการตัดสินใจและค่าความจริงของเงื่อนไข
- 3.2 เขียนคำสั่งเลือกทำแบบเงื่อนไขเดียว สองทาง และหลายทาง
- 3.3 สร้างเงื่อนไขผสมด้วยตัวดำเนินการเปรียบเทียบและตรรกะ
- 3.4 ตรวจสอบเส้นทางการทำงานด้วยชุดข้อมูลทดสอบที่ครอบคลุม

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

- 4.1.1 อธิบาย Boolean expression และการประเมินค่าจริง/เท็จได้
- 4.1.2 เปรียบเทียบการใช้เงื่อนไขแบบทางเลือกเดียว สองทาง และหลายทางได้
- 4.1.3 วิเคราะห์เงื่อนไขซ้อน/เงื่อนไขผสมจากโจทย์ธุรกิจได้

4.2 ทักษะพิสัย

- 4.2.1 แปลง Decision ใน Flowchart เป็นคำสั่งเลือกทำของภาษาที่กำหนดได้
- 4.2.2 เขียนโปรแกรมตัดสินใจหลายเงื่อนไขโดยจัดลำดับเงื่อนไขถูกต้อง
- 4.2.3 สร้างชุดข้อมูลทดสอบเพื่อให้ครอบคลุมทุก branch สำคัญ

4.3 จิตพิสัย

- 4.3.1 ตรวจสอบขอบเขตเงื่อนไขอย่างละเอียด ลดข้อผิดพลาดจากค่าเท่ากัน/ค่าขอบเขต
- 4.3.2 เขียนโค้ดเป็นระเบียบและอธิบายเหตุผลของเงื่อนไขได้

4.4 การประยุกต์ใช้

- 4.4.1 พัฒนาโปรแกรมคำนวณส่วนลดสมาชิก/ระดับลูกค้า หรือคำนวณค่าคอมมิชชั่นตามช่วงยอดขายจากผังงานที่กำหนด

5. สารการเรียนรู้

- 5.1 แนวคิด Boolean และค่าความจริง
- 5.2 ตัวดำเนินการเปรียบเทียบและตรรกะ
- 5.3 คำสั่งเลือกทำแบบเงื่อนไขเดียว (Single Selection)
- 5.4 คำสั่งเลือกทำแบบสองทาง (Double Selection)
- 5.5 คำสั่งเลือกทำแบบหลายทาง (Multiple Selection) และแนวคิด switch/match ตามภาษาที่ใช้
- 5.6 เงื่อนไขซ้อนและลำดับความสำคัญของเงื่อนไข

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ภาศทฤษฎี เวลาเรียน 3 ชั่วโมง

- วิเคราะห์ Decision จาก Flowchart และแปลงเป็น Boolean expression
- ศึกษาโครงสร้างคำสั่งเลือกผ่านตัวอย่างหลายระดับ
- ฝึกหาข้อผิดพลาดจากเงื่อนไขที่ครอบคลุมไม่ครบหรือเรียงลำดับไม่เหมาะสม

6.2 ภาศปฏิบัติ เวลาเรียน 6 ชั่วโมง

- เขียนโปรแกรมแบบ if/else จาก Flowchart
- เพิ่มเงื่อนไขหลายทางและทดสอบค่าขอบเขต
- ทำ peer review เรื่องความครบถ้วนของเงื่อนไขและปรับแก้

6.3 รูปแบบกิจกรรมการเรียนรู้

- ก่อนเรียน แบบทดสอบ/คำถามกระตุ้นความรู้เดิมและชี้แจงภาระงาน
- ระหว่างเรียน อธิบายสั้น-สาธิต-ฝึกตามตัวอย่าง-ฝึกแก้โจทย์-ตรวจทานและรับข้อเสนอแนะ
- หลังเรียน สรุปแนวคิด ส่งชิ้นงาน/หลักฐาน ทำแบบทดสอบหลังเรียน และสะท้อนข้อผิดพลาดที่พบ

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี)

- สไลด์คำสั่งเลือกทำและ Boolean Logic
- Flowchart ที่มี Decision
- IDE/Editor ของรายวิชา
- ชุดโจทย์ธุรกิจที่มีช่วงค่า/ระดับสิทธิ์
- Template Test Case สำหรับตรวจ Branch/Boundary

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 6 ชั่วโมง

ฝึกปฏิบัติจริง

- ใบปฏิบัติงาน พัฒนาโปรแกรม “คำนวณส่วนลดและระดับสมาชิกตามยอดซื้อ”

หลักฐานการปฏิบัติงาน

- Source Code
- Flowchart ที่มี Decision
- Test Case ครอบคลุมทุกระดับสมาชิกและค่าขอบเขต

เกณฑ์การปฏิบัติงาน

- ผลลัพธ์ถูกต้องทุกช่วงเงื่อนไขที่กำหนด
- ไม่มีช่องว่าง/ทับซ้อนของเงื่อนไขที่ทำให้ผลผิด
- โค้ดอ่านง่ายและเส้นทางการตัดสินใจตรงกับ Flowchart

ขอบเขตการปฏิบัติงาน

- พัฒนาโปรแกรมที่มีอย่างน้อย 4 เส้นทางการตัดสินใจ ภายใน 180 นาที และต้องทดสอบค่าขอบเขตอย่างน้อย 2 จุด

วิธีการประเมิน

- สาธิตโปรแกรมด้วย Test Case

- ตรวจ Source Code และ Flowchart เทียบกัน
- สัมภาษณ์เหตุการณ์การเรียงลำดับเงื่อนไข

เครื่องมือประเมิน

- รูบริกส์คำสั่งเลือกทำ
- แบบตรวจ Test Case
- แบบประเมินการสาธิต/สัมภาษณ์

8. การวัดและประเมินผลประจำโมดูล

กำหนดสัดส่วนตามโครงสร้างฝึกเดิม ภาคทฤษฎี 30 คะแนน และภาคปฏิบัติ 70 คะแนน รวม 100 คะแนน

องค์ประกอบการประเมิน	คะแนน	หลักฐาน/เครื่องมือ
แบบทดสอบเงื่อนไขและ Boolean Logic	20	แบบทดสอบ
แบบฝึกวิเคราะห์ Decision/Boundary	10	ใบงาน
ทักษะการเขียนคำสั่งเลือกทำ	40	แบบประเมินปฏิบัติ
ชิ้นงานโปรแกรมส่วนลด/สมาชิก	30	รูบริกส์โปรแกรมและ Test Case
รวม	100	

เกณฑ์พิจารณาผลงาน

- ความถูกต้องตามหลักการ/ผังงานและเงื่อนไขที่กำหนด
- ความสามารถปฏิบัติจริงและอธิบายเหตุผลของวิธีทำ
- คุณภาพหลักฐาน เช่น Source Code Flowchart Test Case Test Report
- ความละเอียด รอบคอบ ความเป็นระเบียบ และการส่งงานตามกำหนด

โมดูลการเรียนรู้ที่ 5 คำสั่งวนซ้ำและการจัดโครงสร้างโปรแกรม

เวลาเรียน 9 ชั่วโมง (ทฤษฎี 3 ชั่วโมง ปฏิบัติ 6 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้ + ทักษะ + จิตพิสัย = การประยุกต์ใช้)

ผู้เรียนเขียนโปรแกรมที่ทำงานซ้ำตามจำนวนครั้งหรือเงื่อนไข เลือกใช้รูปแบบการวนซ้ำเหมาะสม ควบคุมการสิ้นสุดของลูปได้ และจัดโครงสร้างโปรแกรมให้แบ่งงานเป็นส่วนย่อยที่อ่านและทดสอบได้ง่าย

2. อ้างอิงมาตรฐาน

- UoC 10301 ดำเนินการพัฒนาโปรแกรม
- EoC 10301.02 พัฒนาโปรแกรมตามผังงาน (Flow Chart)
- PC อ่านผังงานและลำดับงาน และเขียนโปรแกรมตามผังงาน

3. สมรรถนะประจำโมดูล

- 3.1 อธิบายหลักการทำซ้ำและเงื่อนไขสิ้นสุดของลูป
- 3.2 เขียนลูปแบบนับจำนวนและแบบตรวจเงื่อนไข
- 3.3 ใช้ตัวสะสม ตัวนับ และการประมวลผลข้อมูลหลายรายการ
- 3.4 จัดโปรแกรมเป็นส่วนย่อย/ฟังก์ชันพื้นฐานเพื่อลดความซ้ำซ้อน
- 3.5 ทดสอบลูปทั้งกรณีปกติ กรณีไม่เข้าลูป และกรณีขอบเขต

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

- 4.1.1 อธิบายความแตกต่างระหว่างการวนซ้ำแบบกำหนดจำนวนและแบบอาศัยเงื่อนไขได้
- 4.1.2 วิเคราะห์จำนวนรอบและค่าของตัวนับ/ตัวสะสมจากโค้ดหรือผังงานได้
- 4.1.3 อธิบายประโยชน์ของการแบ่งโปรแกรมเป็นฟังก์ชัน/ส่วนย่อยในระดับพื้นฐานได้

4.2 ทักษะพิสัย

- 4.2.1 เขียนโปรแกรมใช้ Loop ประมวลผลข้อมูลหลายรายการได้
- 4.2.2 ใช้ Counter/Accumulator และตรวจสอบเงื่อนไขสิ้นสุดได้
- 4.2.3 เขียนฟังก์ชันพื้นฐานรับค่า/คืนค่าเพื่อแบ่งงานซ้ำได้
- 4.2.4 ใช้ Trace Table ตรวจสอบการเปลี่ยนแปลงค่าระหว่างรอบได้

4.3 จิตพิสัย

- 4.3.1 ระวัง Infinite Loop และตรวจสอบค่าตั้งต้น/เงื่อนไขสิ้นสุดทุกครั้ง
- 4.3.2 จัดโค้ดเป็นส่วน อ่านง่าย และหลีกเลี่ยงการทำงานซ้ำที่ไม่จำเป็น

4.4 การประยุกต์ใช้

- 4.4.1 พัฒนาโปรแกรมสรุปยอดขายหลายรายการ คำนวณยอดรวม ค่าเฉลี่ย ค่าสูงสุด/ต่ำสุด หรือสรุปจำนวนรายการตามเกณฑ์

5. สารการเรียนรู้

- 5.1 แนวคิด Iteration และ Loop Control
- 5.2 ลูปแบบนับจำนวน (Count-controlled loop)
- 5.3 ลูปแบบเงื่อนไข (Condition-controlled loop)
- 5.4 Counter Accumulator และ Sentinel Value

5.5 ลูปซ้อนในระดับพื้นฐานและข้อควรระวัง

5.6 การแบ่งโปรแกรมเป็นฟังก์ชัน/ส่วนย่อย parameter และ return value เบื้องต้น

5.7 การติดตามค่าด้วย Trace Table และป้องกัน Infinite Loop

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ภาคทฤษฎี เวลาเรียน 3 ชั่วโมง

- เปรียบเทียบ Flowchart ของ Loop รูปแบบต่าง ๆ
- วิเคราะห์ตัวนับ ตัวสะสม และเงื่อนไขหยุดจากตัวอย่าง
- เรียนรู้แนวคิดฟังก์ชันเพื่อแยกงานย่อยและลดโค้ดซ้ำ

6.2 ภาคปฏิบัติ เวลาเรียน 6 ชั่วโมง

- ฝึก Loop จากโจทย์ง่ายสู่โจทย์ประมวลผลหลายรายการ
- เพิ่มฟังก์ชันย่อยสำหรับคำนวณ/แสดงผล
- ทดสอบกรณี 0 รายการ 1 รายการ หลายรายการ และค่าขอบเขต

6.3 รูปแบบกิจกรรมการเรียนรู้

- ก่อนเรียน แบบทดสอบ/คำถามกระตุ้นความรู้เดิมและชี้แจงภาระงาน
- ระหว่างเรียน อธิบายสั้น-สาธิต-ฝึกตามตัวอย่าง-ฝึกแก้โจทย์-ตรวจทานและรับข้อเสนอแนะ
- หลังเรียน สรุปแนวคิด ส่งชิ้นงาน/หลักฐาน ทำแบบทดสอบหลังเรียน และสะท้อนข้อผิดพลาดที่พบ

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี)

- สื่อคำสั่งวนซ้ำและ Trace Table
- Flowchart ที่มี Loop
- IDE/Editor ของรายวิชา
- ชุดข้อมูลตัวอย่างยอดเยี่ยมหลายรายการ
- ใบงานออกแบบฟังก์ชันย่อย

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 6 ชั่วโมง

ฝึกปฏิบัติจริง

- ใบปฏิบัติงาน พัฒนาโปรแกรม “สรุปยอดขายรายวันหลายรายการ”

หลักฐานการปฏิบัติงาน

- Source Code
- Flowchart/Pseudocode ส่วนวนซ้ำ
- Trace Table และผลทดสอบอย่างน้อย 5 กรณี

เกณฑ์การปฏิบัติงาน

- ลูปเริ่มและจบถูกต้อง ไม่มี Infinite Loop
- ยอดรวม/ค่าเฉลี่ย/จำนวนรายการคำนวณถูกต้อง
- แบ่งโค้ดเป็นส่วนย่อยหรือฟังก์ชันตามที่โจทย์กำหนดและอ่านง่าย

ขอบเขตการปฏิบัติงาน

- พัฒนาโปรแกรมรับข้อมูลหลายรายการจนถึงเงื่อนไขสิ้นสุดหรือจำนวนรายการที่กำหนด ภายใน 180 นาที

วิธีการประเมิน

- ตรวจสอบการรันจริงหลาย Test Case
- สังเกตการใช้ Trace Table แก้ปัญหา
- สัมภาษณ์เหตุผลในการเลือกชนิด Loop และการแบ่งฟังก์ชัน

เครื่องมือประเมิน

- แบบประเมินทักษะ Loop/Function
- รุบริกส์ Source Code
- Checklist การทดสอบกรณีขอบเขต

8. การวัดและประเมินผลประจำโมดูล

กำหนดสัดส่วนตามโครงสร้างฝึกเดิม ภาคทฤษฎี 30 คะแนน และภาคปฏิบัติ 70 คะแนน รวม 100 คะแนน

องค์ประกอบการประเมิน	คะแนน	หลักฐาน/เครื่องมือ
แบบทดสอบ Loop/Function	20	แบบทดสอบ
แบบฝึก Trace Table และวิเคราะห์ลูป	10	ใบงาน
ทักษะการเขียน Loop และจัดโครงสร้างโปรแกรม	40	แบบประเมินปฏิบัติ
ชิ้นงานโปรแกรมสรุปยอดขาย	30	รุบริกส์ Source Code/ผลทดสอบ
รวม	100	

เกณฑ์พิจารณาผลงาน

- ความถูกต้องตามหลักการ/ผังงานและเงื่อนไขที่กำหนด
- ความสามารถปฏิบัติจริงและอธิบายเหตุผลของวิธีทำ
- คุณภาพหลักฐาน เช่น Source Code Flowchart Test Case Test Report
- ความละเอียด รอบคอบ ความเป็นระเบียบ และการส่งงานตามกำหนด

โมดูลการเรียนรู้ที่ 6 การพัฒนาโปรแกรมจากผังงานและการทดสอบแบบ Unit Test

เวลาเรียน 9 ชั่วโมง (ทฤษฎี 3 ชั่วโมง ปฏิบัติ 6 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้ + ทักษะ + จิตพิสัย = การประยุกต์ใช้)

ผู้เรียนพัฒนาโปรแกรมขนาดเล็กจาก Work Flow/Flowchart ที่กำหนด ทบทวนแบบก่อนเขียนโค้ด ตรวจสอบจำนวนฟังก์ชันและเงื่อนไขการทดสอบ จัดทำ Test Plan/Unit Test Report และวิเคราะห์ผลเพื่อสรุปความพร้อมใช้งาน พร้อมสาธิตและอธิบายผลงานอย่างเป็นระบบ

2. อ้างอิงมาตรฐาน

- UoC 10301 ดำเนินการพัฒนาโปรแกรม EoC 10301.01 และ 10301.02
- UoC 10302 ทดสอบโปรแกรมแบบ Unit test EoC 10302.01 ดำเนินการ Finish detail design process
- EoC 10302.02 จัดทำรายงาน Unit test เขียน Test plan ระบุ timeline คน resource และวิเคราะห์ผลทดสอบเพื่อ go live หรือ no go live

3. สมรรถนะประจำโมดูล

- 3.1 อ่านและทบทวน Work Flow/Flowchart ก่อนพัฒนาโปรแกรม
- 3.2 พัฒนาโปรแกรมตามผังงานโดยใช้คำสั่งพื้นฐาน การตัดสินใจ การทำซ้ำ และฟังก์ชันตามความเหมาะสม
- 3.3 วิเคราะห์ Functionality และกำหนด Unit/Test Case ที่ต้องทดสอบ
- 3.4 จัดทำ Test Plan ระบุขอบเขต เวลา ผู้รับผิดชอบ/ผู้ทดสอบ และทรัพยากรที่ใช้
- 3.5 บันทึกผล Unit Test วิเคราะห์ข้อผิดพลาด แก้ไข และสรุปผลความพร้อมใช้งาน
- 3.6 สาธิตการทำงานและตอบคำถามเชิงเหตุผลเกี่ยวกับการออกแบบและการทดสอบ

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

- 4.1.1 อธิบายความหมายของ Functionality Unit Test Test Case Expected Result และ Actual Result ได้
- 4.1.2 อธิบายองค์ประกอบของ Test Plan และ Unit Test Report ได้
- 4.1.3 วิเคราะห์ความเชื่อมโยง Requirement/Flowchart/Source Code/Test Case ได้

4.2 ทักษะพิสัย

- 4.2.1 พัฒนาโปรแกรมจาก Flowchart ที่มี Sequence Selection และ Iteration ได้
- 4.2.2 สร้าง Test Case ที่ครอบคลุมฟังก์ชันและเงื่อนไขสำคัญ
- 4.2.3 บันทึกผลทดสอบ แก้ไขข้อผิดพลาด และทดสอบซ้ำ (re-test) ได้
- 4.2.4 จัดทำรายงาน Unit Test พร้อมสรุป go live/no go live ตามผลที่พบ

4.3 จิตพิสัย

- 4.3.1 รับผิดชอบต่อความถูกต้องของผลทดสอบ ไม่ปรับผลให้สวยงามโดยไม่แก้ปัญหามาจริง
- 4.3.2 ทำงานเป็นขั้นตอน เก็บหลักฐาน และสื่อสารข้อจำกัดของโปรแกรมอย่างตรงไปตรงมา

4.4 การประยุกต์ใช้

- 4.4.1 สร้างโปรแกรมธุรกิจขนาดเล็กแบบครบวงจร ตั้งแต่วิเคราะห์ Work Flow/Flowchart เขียนโปรแกรม ทดสอบ Unit Test จนถึงสรุปความพร้อมใช้งาน

5. สารการเรียนรู้

- 5.1 ทบทวนการแปลง Work Flow เป็น Flowchart และ Source Code

- 5.2 การแบ่ง Functionality/หน่วยทดสอบของโปรแกรม
- 5.3 แนวคิด Unit Test และโครงสร้าง Test Case
- 5.4 การกำหนด Test Data Normal Boundary และ Invalid case ตามระดับที่เหมาะสม
- 5.5 องค์ประกอบ Test Plan ขอบเขต Timeline ผู้ทดสอบ/ผู้รับผิดชอบ Resource และเกณฑ์ผ่าน
- 5.6 การจัดทำ Unit Test Report Expected/Actual Result Pass/Fail Defect/Issue Re-test
- 5.7 การสรุปผล go live/no go live จากหลักฐานการทดสอบ
- 5.8 การสัณนิเทศโปรแกรมและการตอบคำถามตามมาตรฐานอาชีพ

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ภาคทฤษฎี เวลาเรียน 3 ชั่วโมง

- ศึกษาองค์ประกอบของ Unit Test และตัวอย่าง Test Plan/Report
- ฝึกแยก Functionality จากโปรแกรมตัวอย่างและเขียน Test Case
- อภิปรายหลักฐานที่เพียงพอต่อการสรุป go live/no go live

6.2 ภาคปฏิบัติ เวลาเรียน 6 ชั่วโมง

- รับโจทย์ Mini Project และจัดทำ Work Flow/Flowchart ฉบับบททวน
- พัฒนา Source Code ตามผังงานและตรวจการทำงานที่ละ Functionality
- จัดทำ Test Plan รัน Unit Test บันทึกผล แก้ไข และ Re-test
- จัดทำ Unit Test Report และสัณนิเทศโปรแกรมพร้อมสัมภาษณ์

6.3 รูปแบบกิจกรรมการเรียนรู้

- ก่อนเรียน แบบทดสอบ/คำถามกระตุ้นความรู้เดิมและชี้แจงภาระงาน
- ระหว่างเรียน อธิบายสั้น-สัณนิเทศ-ฝึกตามตัวอย่าง-ฝึกแก้โจทย์-ตรวจทานและรับข้อเสนอแนะ
- หลังเรียน สรุปแนวคิด ส่งชิ้นงาน/หลักฐาน ทำแบบทดสอบหลังเรียน และสะท้อนข้อผิดพลาดที่พบ

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี)

- โจทย์ Mini Project งานธุรกิจขนาดเล็ก
- Template Flowchart/Pseudocode/Source Code Checklist
- Template Test Plan Test Case และ Unit Test Report
- IDE/Editor และเครื่องมือรันภาษาคอมพิวเตอร์ที่สถานศึกษากำหนด
- ตัวอย่าง Rubric การพัฒนาและทดสอบโปรแกรม

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 6 ชั่วโมง

ฝึกปฏิบัติจริง

- ใบปฏิบัติงาน Mini Project “ระบบคำนวณค่าบริการ/ยอดชำระสำหรับงานธุรกิจ” พร้อม Unit Test

หลักฐานการปฏิบัติงาน

- Work Flow/Flowchart และ Pseudocode ฉบับสมบูรณ์
- Source Code ที่รันได้
- Test Plan ที่ระบุ timeline ผู้ทดสอบ/ผู้รับผิดชอบ และ resource
- ชุด Test Case และ Unit Test Report พร้อมหลักฐานผลรัน

- สรุป go live/no go live และรายการข้อจำกัด/สิ่งที่ต้องแก้ไข

เกณฑ์การปฏิบัติงาน

- Source Code สอดคล้องกับ Flowchart และ Functionality ที่กำหนด
- Test Case ครอบคลุมฟังก์ชันและเงื่อนไขสำคัญ โดยมี Expected/Actual Result ชัดเจน
- กรณี Fail มีการวิเคราะห์ แก้ไข และ Re-test พร้อมหลักฐาน
- การสรุป go live/no go live อ้างอิงผลทดสอบ ไม่ใช่ความเห็นลอย ๆ
- ผู้เรียนสาธิตและอธิบายกระบวนการพัฒนา/ทดสอบได้

ขอบเขตการปฏิบัติงาน

- พัฒนา Mini Project 1 โปรแกรม โดยผู้สอนกำหนดขอบเขตประมาณ 3-5 Functionality ใช้เวลาในชั้น 180 นาทีสำหรับการพัฒนา/ทดสอบหลัก และส่งเอกสารประกอบตามกำหนด

วิธีการประเมิน

- ข้อสอบข้อเขียนเรื่อง Unit Test/Test Plan
- การสาธิตโปรแกรมและการปฏิบัติงาน
- การสัมภาษณ์การอ่าน Flowchart/Source Code/Test Result
- ประเมินแฟ้มหลักฐานและ Unit Test Report

เครื่องมือประเมิน

- แบบทดสอบความรู้
- รุบริกส์ Mini Project
- แบบประเมินการสาธิตและสัมภาษณ์
- Checklist Test Plan/Test Case/Unit Test Report

8. การวัดและประเมินผลประจำโมดูล

กำหนดสัดส่วนตามโครงสร้างฝึกเดิม ภาคทฤษฎี 30 คะแนน และภาคปฏิบัติ 70 คะแนน รวม 100 คะแนน

องค์ประกอบการประเมิน	คะแนน	หลักฐาน/เครื่องมือ
แบบทดสอบ Unit Test/Test Plan	20	แบบทดสอบ
แบบฝึกแยก Functionality และออกแบบ Test Case	10	ใบงาน
กระบวนการพัฒนาและทดสอบโปรแกรม	40	แบบประเมินปฏิบัติ/สาธิต
Mini Project + Unit Test Report	30	รุบริกส์ชิ้นงานและแฟ้มหลักฐาน
รวม	100	

เกณฑ์พิจารณาผลงาน

- ความถูกต้องตามหลักการ/ผังงานและเงื่อนไขที่กำหนด
- ความสามารถปฏิบัติงานจริงและอธิบายเหตุผลของวิธีทำ
- คุณภาพหลักฐาน เช่น Source Code Flowchart Test Case Test Report
- ความละเอียด รอบคอบ ความเป็นระเบียบ และการส่งงานตามกำหนด

การประเมินผลลัพธ์การเรียนรู้ปลายรายวิชา

เวลา 3 ชั่วโมง (ทฤษฎี 1 ชั่วโมง ปฏิบัติ 2 ชั่วโมง)

วัตถุประสงค์การประเมิน

- ประเมินความเข้าใจหลักการเขียนโปรแกรม การวิเคราะห์ปัญหา Algorithm Flowchart และ Pseudocode
- ประเมินความสามารถอ่านผังงานและพัฒนาโปรแกรมตามผังงาน
- ประเมินความสามารถทดสอบโปรแกรม จัดทำ Test Case/Unit Test Report และอธิบายผล
- ประเมินความละเอียด รอบคอบ ความเป็นระบบ และการสื่อสารผลงาน

ภาระงานประเมินบูรณาการ

ผู้เรียนรับโจทย์ธุรกิจขนาดเล็ก 1 สถานการณ์ แล้วดำเนินการตั้งแต่วิเคราะห์ข้อกำหนด จัดทำ Flowchart/Pseudocode พัฒนาโปรแกรม ทดสอบ และส่งหลักฐาน พร้อมสาธิตโปรแกรมและตอบคำถามตามเวลาที่กำหนด

หลักฐานที่ต้องส่ง

- ใบวิเคราะห์ข้อกำหนดและ Input-Process-Output
- Flowchart และ Pseudocode
- Source Code
- Test Case/Unit Test Report
- หลักฐานผลการรันและสรุปข้อผิดพลาด/การแก้ไข
- สรุปผลความพร้อมใช้งาน (go live/no go live) ตามหลักฐานการทดสอบ

รูบริกการประเมินบูรณาการ (100 คะแนน)

รายการประเมิน	คะแนน	แนวพิจารณา
การวิเคราะห์โจทย์และ Flowchart/Pseudocode	20	ข้อกำหนดครบ ลำดับงานถูกต้อง สอดคล้องกัน
ความถูกต้องของโปรแกรม	30	ทำงานตามโจทย์และผังงาน ผลลัพธ์ถูกต้อง
โครงสร้างและความอ่านง่ายของ Source Code	15	ชื่อตัวแปร/การจัดรูปแบบ/การแบ่งส่วนเหมาะสม
Test Plan/Test Case/Unit Test Report	20	ครอบคลุม Functionality มี Expected/Actual และหลักฐาน
การสาธิตและสัมภาษณ์	10	รันจริง อธิบายเหตุผล อ่านผังงาน/โค้ด/ผลทดสอบได้
กิกินัยและความละเอียดรอบคอบ	5	ตรวจทาน เก็บหลักฐาน เป็นระเบียบ รับผิดชอบ

หมายเหตุ เกณฑ์ผ่าน/การแปลงคะแนนเป็นผลการเรียนให้ยึดระเบียบการวัดและประเมินผลของสถานศึกษา

โดยผู้สอนสามารถปรับสัดส่วนคะแนนย่อยให้เหมาะสมกับบริบทชั้นเรียนได้

ภาคผนวก การอ้างอิงมาตรฐานอาชีพที่ใช้ในการออกแบบโมดูล

UoC	EoC	คำอธิบาย	เกณฑ์การปฏิบัติงาน (PC) ที่นำมาใช้	วิธีประเมินตามมาตรฐาน
10301	10301.01	แปลงหน้าจอ (GUI) หรือลำดับงาน (Work Flow) เป็นผังงาน (Flow Chart)	อ่านแบบร่าง/Work Flow ทบทวนและแก้ไข เขียน Flow Chart ตามที่กำหนด	ข้อสอบข้อเขียน การสัมภาษณ์ การสาธิตการปฏิบัติงาน
10301	10301.02	พัฒนาโปรแกรมตามผังงาน (Flow Chart)	อ่านผังงานและลำดับงาน เขียนโปรแกรมตามผังงาน	ข้อสอบข้อเขียน การสัมภาษณ์ การสาธิตการปฏิบัติงาน
10302	10302.01	ดำเนินการ Finish detail design process	วิเคราะห์จำนวน Functionality ตรวจสอบเงื่อนไขการทดสอบ	ข้อสอบข้อเขียน การสาธิตการปฏิบัติงาน
10302	10302.02	จัดทำรายงาน Unit test	เขียน Test plan ระบุ timeline คน resource วิเคราะห์ผลทดสอบเพื่อ go live หรือ no go live	ข้อสอบข้อเขียน การสาธิตการปฏิบัติงาน