



## แผนการจัดการเรียนรู่มุ่งเน้นสมรรถนะ

ชื่อวิชา การเขียนโปรแกรมคอมพิวเตอร์

รหัสวิชา 20105-2118 ทฤษฎี 1 ปฏิบัติ 3 หน่วยกิต 2

หลักสูตรประกาศนียบัตรวิชาชีพ  หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง

ประเภทวิชา ช่างอุตสาหกรรม

สาขาวิชา อิเล็กทรอนิกส์

จัดทำโดย

นายเกียรติศักดิ์ สุวรรณบุตร

วิทยาลัยการอาชีพบ้านฝื่อ

สำนักงานคณะกรรมการการอาชีวศึกษา

กระทรวงศึกษาธิการ



## หลักสูตรรายวิชา

ชื่อวิชา การเขียนโปรแกรมคอมพิวเตอร์

รหัสวิชา 20105-2118 ทฤษฎี 1 ปฏิบัติ 3 หน่วยกิต 2

หลักสูตรประกาศนียบัตรวิชาชีพ

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง

ประเภทวิชา ช่างอุตสาหกรรม

สาขาวิชา อิเล็กทรอนิกส์

### จุดประสงค์รายวิชา

1. เข้าใจหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์
2. มีทักษะในการวางแผน เขียน ตรวจสอบ และแก้ไขโปรแกรมคอมพิวเตอร์
3. มีเจตคติที่ดีต่อวิชาชีพ มีกึ๋นสั้ยในการค้นคว้าเพิ่มเติม ปฏิบัติงานด้วยความละเอียดรอบคอบ คำนึงถึงความปลอดภัย

ถึงความถูกต้องปลอดภัย

### สมรรถนะรายวิชา

1. แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรมภาษาอื่น ๆ
2. เขียนโปรแกรมด้วยคอมพิวเตอร์

### คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการของภาษา องค์ประกอบของโปรแกรม คำสั่ง ตัวแปร ฟิลชาร์ต ฟังก์ชันโปรแกรมย่อย การวางแผนและเขียนโปรแกรม ตรวจสอบ แก้ไขโปรแกรม โดยเลือกใช้โปรแกรมภาษาซี หรือโปรแกรมภาษาอื่น ๆ

## ตารางวิเคราะห์หลักสูตร

รหัสวิชา 20105-2118 วิชา การเขียนโปรแกรมคอมพิวเตอร์ จำนวน 2 หน่วยกิต  
 ชั้น ปวช.2 สาขาวิชา/กลุ่มวิชา/ อิเล็กทรอนิกส์

เรียนรู้อะไร ชื่อหน่วยการสอน/การเรียนรู้	ด้านพุทธิพิสัย						ด้านทักษะพิสัย (5)	ด้านจิตพิสัย (5)	รวม (40)	ลำดับความสำคัญ	จำนวนชั่วโมง
	ความรู้ (5)	ความเข้าใจ (5)	นำไปใช้ (5)	วิเคราะห์ (5)	สังเคราะห์ (5)	ประเมินค่า (5)					
1. องค์ประกอบของคอมพิวเตอร์และโปรแกรมภาษาซี	5	5	5	4	5	5	5	5	39	2	8
2. โปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	5	5	5	5	4	5	5	5	39	2	8
3. การเขียนโปรแกรมเบื้องต้นด้วยผังงาน	5	5	5	5	5	5	5	5	40	1	8
4. การแสดงผลข้อมูลและการรับค่า	5	5	5	4	4	5	5	5	38	3	8
5. ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	5	5	5	5	5	4	5	5	39	2	8
6. ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	5	5	5	5	5	5	4	5	39	2	4
7. ฟังก์ชันการทำงานแบบทำซ้ำ	5	5	5	5	5	4	5	5	39	2	4
8. ฟังก์ชันภาษาซี	5	5	5	5	4	5	4	5	38	3	8
9. อาร์เรย์	5	5	4	5	5	5	5	5	39	2	4
10. พอยน์เตอร์	5	5	4	4	5	4	5	5	37	5	4
11. ข้อมูลชนิดโครงสร้างและยูเนียน	5	5	4	5	5	4	5	5	38	4	4
12. ภาษาซีกับไฟล์ข้อมูล	5	5	4	4	5	4	5	5	37	5	4
<b>รวมคะแนน</b>											
<b>ลำดับความสำคัญ</b>											

คำอธิบาย 5 หมายถึง ระดับความสำคัญของแต่ละรายการมี 5 ระดับ คือ 1, 2, 3, 4, 5

## หน่วยการเรียนรู้

หน่วยที่	ชื่อหน่วยการเรียนรู้	จำนวน ชั่วโมง	สัปดาห์ที่
1	องค์ประกอบของคอมพิวเตอร์และโปรแกรมภาษาซี	8	1-2
2	โปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	8	3-4
3	การเขียนโปรแกรมเบื้องต้นด้วยผังงาน	8	5-6
4	การแสดงผลข้อมูลและการรับค่า	8	7-8
5	ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	8	9-10
6	ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	4	11
7	ฟังก์ชันการทำงานแบบทำซ้ำ	4	12
8	ฟังก์ชันภาษาซี	8	13-14
9	อาร์เรย์	4	15
10	พอยน์เตอร์	4	16
11	ข้อมูลชนิดโครงสร้างและยูเนียน	4	17
12	ภาษาซีกับไฟล์ข้อมูล	4	18

### หน่วยการเรียนรู้และสมรรถนะประจำหน่วย

ชื่อหน่วย	สมรรถนะ		
	ความรู้	ทักษะ	คุณลักษณะที่พึงประสงค์
1. องค์ประกอบของคอมพิวเตอร์และโปรแกรมภาษาซี	-แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี	-เขียนโปรแกรมภาษาซีได้ -แสดงผลของการเขียนภาษาซีได้	1.มีวินัย 2.ใฝ่เรียนรู้ 3.มีเหตุผล
2. โปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	-แสดงความรู้เกี่ยวกับโปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	-เขียนโปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี -แสดงผลของการเขียนโปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	1.มีวินัย 2.ใฝ่เรียนรู้ 3.รู้จักกาลเทศะ
3. การเขียนโปรแกรมเบื้องต้นด้วยผังงาน	-แสดงความรู้เกี่ยวกับการเขียนโปรแกรมเบื้องต้นด้วยผังงาน	-เขียนโปรแกรมเบื้องต้นด้วยผังงาน -แสดงผลของการเขียนโปรแกรมเบื้องต้นด้วยผังงาน	1.มีวินัย 2.ใฝ่เรียนรู้ 3.มีเหตุผล
4. การแสดงผลข้อมูลและการรับค่า	-แสดงความรู้เกี่ยวกับการแสดงผลข้อมูลและการรับค่า	-เขียนโปรแกรมการแสดงผลข้อมูลและการรับค่า -แสดงผลของการเขียนการแสดงผลข้อมูลและการรับค่า	1.มีวินัย 2.ใฝ่เรียนรู้ 3.รู้จักกาลเทศะ

5. ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	-แสดงความรู้เกี่ยวกับตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	-เขียนโปรแกรมตัวดำเนินการและนิพจน์ทางคณิตศาสตร์ -แสดงผลของการพจน์ทางคณิตศาสตร์	1. มีวินัย 2. ใฝ่เรียนรู้ 3. รู้จักกาลเทศะ
6. ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	-แสดงความรู้เกี่ยวกับฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	-เขียนฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก -แสดงผลเกี่ยวกับฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	1. มีวินัย 2. ใฝ่เรียนรู้ 3. มีเหตุผล
7. ฟังก์ชันการทำงานแบบทำซ้ำ	-แสดงความรู้เกี่ยวกับฟังก์ชันการทำงานแบบทำซ้ำ	-เขียนฟังก์ชันการทำงานแบบทำซ้ำ -แสดงผลการเขียนฟังก์ชันการทำงานแบบทำซ้ำ	1. มีวินัย 2. ใฝ่เรียนรู้ 3. รู้จักกาลเทศะ
8. ฟังก์ชันภาษาซี	-แสดงความรู้เกี่ยวกับฟังก์ชันภาษาซี	-เขียนและเรียนรู้ฟังก์ชันภาษาซี -แสดงผลการเขียน ภาษาซี	1. มีวินัย 2. ใฝ่เรียนรู้ 3. มีเหตุผล
9. อาร์เรย์	-แสดงความรู้เกี่ยวกับ อาร์เรย์	-เขียนและเรียนรู้เกี่ยวกับอาร์เรย์ -แสดงผลการเขียน อาร์เรย์	1. มีวินัย 2. ใฝ่เรียนรู้ 3. รู้จักกาลเทศะ
10. พอยน์เตอร์	-แสดงความรู้เกี่ยวกับพอยน์เตอร์	-เขียนและเรียนรู้เกี่ยวกับพอยน์เตอร์ -แสดงผลการเขียน พอยน์เตอร์	1. มีวินัย 2. ใฝ่เรียนรู้ 3. มีเหตุผล

11. ข้อมูลชนิดโครงสร้างและยูเนียน	-แสดงความรู้เกี่ยวกับข้อมูลชนิดโครงสร้างและยูเนียน	-เขียนและเรียนรู้เกี่ยวกับข้อมูลชนิดโครงสร้างและยูเนียน -แสดงผลการเขียน ข้อมูลชนิดโครงสร้างและยูเนียน	1.มีวินัย 2.ใฝ่เรียนรู้ 3.มีเหตุผล
12. ภาษาซีกับไฟล์ข้อมูล	-แสดงความรู้เกี่ยวกับภาษาซีกับไฟล์ข้อมูล	-เขียนและเรียนรู้เกี่ยวกับภาษาซีกับไฟล์ข้อมูล -แสดงผลการเขียน ภาษาซีกับไฟล์ข้อมูล	1.มีวินัย 2.ใฝ่เรียนรู้ 3.มีเหตุผล

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>1. สาระสำคัญ</b></p> <p>ศึกษาหลักการและขั้นตอนการพัฒนาโปรแกรมคอมพิวเตอร์ เครื่องมือการออกแบบโปรแกรม ความ เป็นมาของภาษาคอมพิวเตอร์ โครงสร้างภาษาคอมพิวเตอร์ ลักษณะเด่นและด้อยของ ภาษาคอมพิวเตอร์ ประเภทข้อมูล องค์ประกอบของคำสั่ง การคำนวณและเปรียบเทียบ ขั้นตอนในการ ทำงานของโอเปอเรชันคำนวณ คำสั่งต่างๆ ในภาษาคอมพิวเตอร์ ฟังก์ชันเบื้องต้น การเขียนโปรแกรมงาน ต่างๆ โดยใช้คำสั่งและฟังก์ชันเบื้องต้นในภาษาใดภาษาหนึ่งกับเครื่องมือคอมพิวเตอร์ ปฏิบัติการ วิเคราะห์โจทย์ปัญหา ออกแบบโปรแกรมและเขียนโปรแกรมด้วยคำสั่งภาษาคอมพิวเตอร์ เพื่อให้มีความรู้ ความเข้าใจ และทักษะในการวิเคราะห์โจทย์ปัญหา สามารถออกแบบโปรแกรมและเขียน โปรแกรมใช้งาน รวมทั้งสามารถนำไปประยุกต์ใช้งานได้</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. บอกภาษาคอมพิวเตอร์ที่รู้จักได้</p> <p>2.2. อธิบายลักษณะต่าง ๆ ของโปรแกรมแปลภาษาคอมพิวเตอร์ได้</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 บอกภาษาคอมพิวเตอร์ที่รู้จักได้</p> <p>3.1.2 อธิบายลักษณะต่าง ๆ ของโปรแกรมแปลภาษาคอมพิวเตอร์ได้</p> <p>3.1.3 อธิบายลักษณะต่าง ๆ ของโปรแกรมแปลภาษาคอมพิวเตอร์ได้</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีวินัย</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 อยู่อย่างพอเพียง</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ประวัติความเป็นมาของคอมพิวเตอร์

คอมพิวเตอร์ที่ใช้อยู่ทุกวันนี้เป็นผลมาจากการประดิษฐ์คิดค้นเครื่องมือในการคำนวณ ซึ่งมีวิวัฒนาการ เริ่มจากเครื่องมือในการคำนวณเครื่องแรก คือ “ลูกคิด (Abacus)” ที่สร้างขึ้นในประเทศจีนเมื่อประมาณ 2,000-3,000 ปีมาแล้ว จนกระทั่งในปี พ.ศ. 2376 นักคณิตศาสตร์ชาวอังกฤษชื่อชาร์ล แบบเบจ (Charles Babbage) ได้ประดิษฐ์เครื่องวิเคราะห์ (Analytical Engine) ที่สามารถคำนวณค่าของฟังก์ชันตรีโกณมิติ และฟังก์ชันต่างๆ ทางคณิตศาสตร์ หลักการของแบบเบจนี้เองที่ได้นำมาพัฒนาสร้างเครื่องคอมพิวเตอร์ สมัยใหม่แบบเบจจึงถูกยกย่องให้เป็นบิดาแห่งเครื่องคอมพิวเตอร์ หลังจากนั้นเป็นต้นมา ได้มีผู้ประดิษฐ์ เครื่องคอมพิวเตอร์ขึ้นมามากมายหลายขนาด ซึ่งถือว่าการเริ่มยุคของคอมพิวเตอร์อย่างแท้จริง โดยสามารถจัดแบ่งคอมพิวเตอร์ออกได้เป็น 5 ยุค ดังนี้

##### 1.2 อันตรรกของไฟฟ้าต่อร่างกายมนุษย์

**1.1 ยุคที่หนึ่ง (First Generation Computer) พ.ศ. 2489-2501** เป็นการประดิษฐ์เครื่องคอมพิวเตอร์ที่ไม่ใช่เครื่องคำนวณ โดยเมาชลีและเอ็กเคอร์ต (Mauchly and Eckert) ได้นำแนวความคิดมาประดิษฐ์เป็นเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพมากเครื่องหนึ่งเรียกว่า ENIAC (Electronic Numerical Integrator and Calculator) ซึ่งต่อมาได้ปรับปรุงการทำงานของเครื่องคอมพิวเตอร์ให้มีประสิทธิภาพดียิ่งขึ้น และได้ประดิษฐ์เครื่อง UNIVAC (Universal Automatic Computer) ขึ้นเพื่อใช้ในการสำรวจสำมะโนประชากรประจำปี จึงนับได้ว่า UNIVAC เป็นเครื่องคอมพิวเตอร์ เครื่องแรกของโลกที่ถูกใช้งานในเชิงธุรกิจ ซึ่งนับว่าเป็นการเริ่มต้นของเครื่องคอมพิวเตอร์ในยุคแรกอย่างแท้จริง เครื่องคอมพิวเตอร์ในยุคนี้ใช้หลอดสุญญากาศในการควบคุมการทำงานของเครื่องซึ่งทำงานได้อย่างรวดเร็ว แต่มีขนาดใหญ่มากและราคาแพง ยุคแรกของคอมพิวเตอร์สิ้นสุดเมื่อมีผู้ประดิษฐ์ทรานซิสเตอร์มาใช้แทน หลอดสุญญากาศ



รูปที่ 1.1 เครื่องคอมพิวเตอร์ในยุคที่ใช้หลอดสุญญากาศในการควบคุม

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

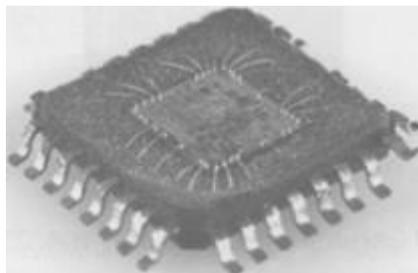
**1.2 ยุคที่สอง (Second Generation Computer) พ.ศ. 2502-2506** มีการนำทรานซิสเตอร์ มาใช้ในเครื่องคอมพิวเตอร์จึงทำให้เครื่องมีขนาดเล็กลง และสามารถเพิ่มประสิทธิภาพในการทำงานให้มีความรวดเร็วและแม่นยำมากยิ่งขึ้น นอกจากนี้ยังได้มีการคิดภาษาเพื่อใช้กับเครื่องคอมพิวเตอร์ เช่น ภาษาฟอร์แทรน (FORTRAN) จึงทำให้ง่ายต่อการเขียนโปรแกรมเพื่อใช้กับเครื่อง ในยุคนี้ใช้อุปกรณ์ทรานซิสเตอร์ (Transistor) ซึ่งสร้างจากสารกึ่งตัวนำ (Semi-conductor) เป็นอุปกรณ์หลักแทนหลอดสุญญากาศ เนื่องจากทรานซิสเตอร์เพียงตัวเดียวมีประสิทธิภาพในการทำงานเทียบเท่าหลอดสุญญากาศได้นับร้อย หลอดทำให้เครื่องคอมพิวเตอร์ในยุคนี้มีขนาดเล็ก ใช้พลังงานไฟฟ้าน้อยความร้อนต่ำทำงานเร็ว และได้รับ ความน่าเชื่อถือมากยิ่งขึ้น เก็บข้อมูลได้โดยใช้ส่วนความจำวงแหวนแม่เหล็ก (Magnetic Core) มีความเร็ว ในการประมวลผลในหนึ่งคำสั่ง ประมาณหนึ่งในพันของวินาที (Ms: Millisecond) สั่งงานได้สะดวก มากขึ้น เนื่องจากทำงานด้วยภาษาสัญลักษณ์ (Assembly Language) เริ่มพัฒนาภาษาระดับสูง (High Level Language) ขึ้นใช้งานในยุคนี้



รูปที่ 1.2 การเขียนโปรแกรมสำหรับใช้กับเครื่องใช้อุปกรณ์ทรานซิสเตอร์

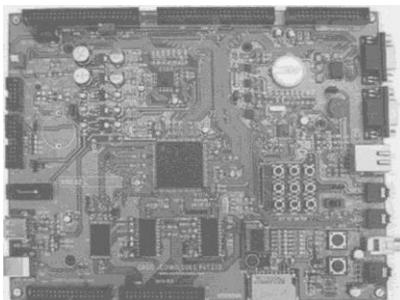
**1.3 ยุคที่สาม (Third Generation Computer) พ.ศ. 2507-2512** คอมพิวเตอร์ในยุคนี้ เริ่มต้นภายหลังจากการใช้ทรานซิสเตอร์ได้เพียง 5 ปีเนื่องจากได้มีการประดิษฐ์คิดค้นเกี่ยวกับวงจรรวม (Integrated Circuit) หรือเรียกกันย่อๆ ว่า “ไอซี (IC)” ซึ่งไอซีนี้ทำให้ส่วนประกอบและวงจรต่างๆ สามารถวางลงได้บนแผ่นชิป (Chip) เล็กๆ เพียงแผ่นเดียว จึงมีการนำเอาแผ่นชิปมาใช้แทนทรานซิสเตอร์ทำให้ประหยัดเนื้อที่ได้ มาก นอกจากนี้ยังเริ่มมีการใช้งานระบบจัดการฐานข้อมูล (DBMS: Data Base Management Systems) และมีการพัฒนาเครื่องคอมพิวเตอร์ให้สามารถทำงานร่วมกันได้หลายๆ งานในเวลาเดียวกัน และมีระบบที่ ผู้ใช้สามารถโต้ตอบกับเครื่องได้หลายๆ เครื่องพร้อมกัน (Time Sharing)

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8



รูปที่ 1.3 วงจรรวม (Integrated Circuit)

**1.4 ยุคที่สี่ (Fourth Generation Computer) พ.ศ. 2513-2532** เป็นยุคที่นำสารกึ่งตัวนำมาสร้างเป็นวงจรรวมความจุสูงมาก (VLSI: Very Large Scale integrated) ซึ่งสามารถย่อส่วนไอซี ธรรมดาหลายๆ วงจรเข้ามาในวงจรเดียวกัน มีการประดิษฐ์ไมโครโพรเซสเซอร์ (Microprocessor) ขึ้น ทำให้เครื่องมีขนาดเล็ก ราคาถูกลง และมีความสามารถในการทำงานสูงและรวดเร็วมาก จึงทำให้มีคอมพิวเตอร์ส่วนบุคคล (Personal Computer) ถือกำเนิดขึ้นมาในยุคนี้



รูปที่ 1.4 วงจรรวมความจุสูงมาก (VLSI: Very Large Scale Integrated)

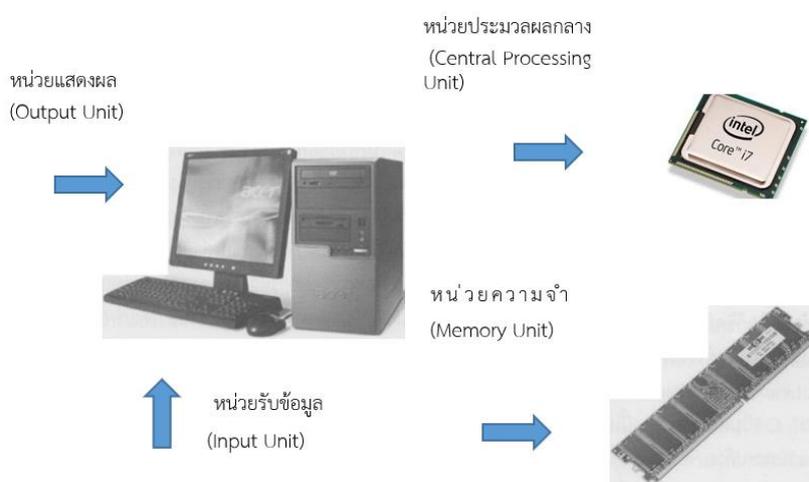
**1.5 ยุคที่ห้า (Fifth Generation Computer) พ.ศ. 2533 จนถึงปัจจุบัน ในยุคนี้ได้** มุ่งเน้นการพัฒนาความสามารถในการทำงานของระบบคอมพิวเตอร์และความสะดวกสบายในการใช้งาน เครื่องคอมพิวเตอร์อย่างชัดเจน มีการพัฒนาสร้างเครื่องคอมพิวเตอร์แบบพกพาขนาดเล็ก (Portable Computer) ขึ้นใช้งานในโครงการพัฒนาอุปกรณ์ VLSI ให้ใช้งานง่ายและมีความสามารถสูงขึ้น รวมทั้ง โครงการวิจัยและพัฒนาเกี่ยวกับปัญญาประดิษฐ์ (AI: Artificial Intelligence) เป็นหัวใจของการพัฒนา ระบบคอมพิวเตอร์ในยุคนี้โดยทำให้ระบบคอมพิวเตอร์มีความรู้สามารถวิเคราะห์ปัญหาด้วยเหตุผล

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่	1-2
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

## 2. ส่วนประกอบของคอมพิวเตอร์เบื้องต้น

เครื่องคอมพิวเตอร์เป็นเพียงองค์ประกอบส่วนหนึ่งของระบบคอมพิวเตอร์เท่านั้น การทำให้เครื่องคอมพิวเตอร์แต่ละเครื่องสามารถทำงานได้อย่างมีประสิทธิภาพตามที่ต้องการนั้น จำเป็นต้องอาศัยองค์ประกอบพื้นฐานมาทำงานร่วมกัน ซึ่งองค์ประกอบพื้นฐานของระบบคอมพิวเตอร์ประกอบไปด้วย

**2.1 ฮาร์ดแวร์ (Hardware)** หมายถึง อุปกรณ์ต่างๆ ที่ประกอบขึ้นเป็นเครื่องคอมพิวเตอร์ มีลักษณะเป็นโครงร่างสามารถมองเห็นด้วยตาและสัมผัสได้ เช่น จอภาพ คีย์บอร์ด เครื่องพิมพ์ เมาส์ เป็นต้น ซึ่งสามารถแบ่งออกเป็นส่วนต่างๆ ตามลักษณะการทำงานได้ 4 หน่วย คือ หน่วยรับข้อมูล (Input Unit) หน่วยประมวลผลกลาง (CPU: Central Processing Unit) หน่วยแสดงผล (Output Unit) และ หน่วยความจำ (Memory) โดยอุปกรณ์แต่ละหน่วยมีหน้าที่การทำงานแตกต่างกัน



รูปที่ 1.5 อุปกรณ์ต่างๆ ที่ประกอบขึ้นเป็นเครื่องคอมพิวเตอร์

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>2.1.1 หน่วยประมวลผลกลาง (CPU: Central Processing Unit) เป็นศูนย์กลาง การประมวลผลของทั้งระบบ เปรียบเสมือนกองบัญชาการหรือส่วนของศีรษะของมนุษย์ที่มีผู้บัญชาการ หรือสมอง อยู่ภายใน</b></p> <p><b>2.1.2 หน่วยความจำ (Memory Unit) ทำหน้าที่เก็บชุดคำสั่งหรือข้อมูลในระหว่าง การประมวลผลหรือที่ได้จากการประมวลผลแบ่งได้หลักๆ ดังนี้</b></p> <p>1) หน่วยความจำหลัก (Main Memory) เป็นส่วนความจำพื้นฐานในคอมพิวเตอร์ ทุกเครื่อง เป็นหัวใจของการทำงานในรูปแบบอัตโนมัติ มีหน้าที่เก็บข้อมูลต่างๆ ที่ป้อนเข้ามาเพื่อให้ส่วนประมวลผลนำไปใช้ และเก็บข้อมูลที่เกี่ยวข้องกันคุณสมบัติและระบบการทำงานของเครื่องคอมพิวเตอร์ แบ่งเป็น 2 ประเภท คือ</p> <p>(1) Random Access Memory (RAM) เรียกอีกอย่างว่าหน่วยความจำชั่วคราว อันเนื่องมาจากลักษณะการทำงานของ RAM จะจำข้อมูลได้เฉพาะเวลาที่มีกระแสไฟฟ้าหล่อเลี้ยงเท่านั้น</p> <p>(2) Read Only Memory (ROM) เรียกอีกอย่างว่าหน่วยความจำถาวร คือ ถึงแม้ไม่มีกระแสไฟฟ้าหล่อเลี้ยงก็ยังจำข้อมูลอยู่</p> <p>2) หน่วยความจำสำรอง (Secondary storage) เป็นหน่วยความจำที่ใช้ในการเก็บชุดคำสั่งหรือข้อมูลไว้</p> <p><b>2.1.3 หน่วยแสดงผล (OU: Output Unit) ทำหน้าที่ในการแสดงผลจากคอมพิวเตอร์ เพื่อสื่อสารให้มนุษย์เข้าใจได้</b></p> <p><b>2.1.4 หน่วยรับคำสั่งหรือข้อมูล (IU: Input Unit) เป็นส่วนที่ทำหน้าที่นำข้อมูลจาก ภายนอกเข้าสู่เครื่องคอมพิวเตอร์ เสมือนเป็นประสาทสัมผัสของคอมพิวเตอร์ในการรับคำสั่งหรือข้อมูลเข้า คอมพิวเตอร์เพื่อประมวลผลต่อไป เช่น เมาส์ คีย์บอร์ด เป็นต้น</b></p> <p><b>2.2 ซอฟต์แวร์ (Software) เป็นโปรแกรมหรือชุดคำสั่งที่ถูกเขียนขึ้นเพื่อสั่งให้เครื่องคอมพิวเตอร์ทำงานซอฟต์แวร์จึงเป็นเหมือนตัวเชื่อมระหว่างผู้ใช้กับเครื่องคอมพิวเตอร์ถ้าไม่มีซอฟต์แวร์ก็ไม่สามารถใช้เครื่องคอมพิวเตอร์ทำอะไรได้เลย ซอฟต์แวร์สำหรับเครื่องคอมพิวเตอร์สามารถแบ่งได้ ดังนี้</b></p> <p><b>2.2.1 ซอฟต์แวร์ระบบ (System Software) คือ ชุดของคำสั่งที่เขียนไว้เป็นคำสั่งสำเร็จรูป ซึ่งจะทำงานใกล้ชิดกับคอมพิวเตอร์มากที่สุด เพื่อคอยควบคุมการทำงานของฮาร์ดแวร์ทุกอย่าง และอำนวยความสะดวกให้กับผู้ใช้ในการใช้งาน ซอฟต์แวร์หรือโปรแกรมระบบที่รู้จักกันดี ได้แก่ DOS, Windows, UNIX,</b></p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p>Linux รวมทั้งโปรแกรมแปลคำสั่งที่เขียนในภาษาระดับสูง เช่น ภาษา Basic, FORTRAN, Pascal, COBOL, c เป็นต้น นอกจากนี้โปรแกรมที่ใช้ในการตรวจสอบระบบ เช่น Norton's utilities ก็นับเป็น ซอฟต์แวร์ระบบด้วย</p> <p style="text-align: center;"><b>2.2.2 ซอฟต์แวร์ประยุกต์ (Application Software)</b> คือ ซอฟต์แวร์หรือโปรแกรมที่ส่งคอมพิวเตอร์ทำงานต่างๆ ตามที่ผู้ใช้งานต้องการไม่ว่าจะเป็นด้านเอกสาร บัญชี หรือการจัดเก็บข้อมูล ซอฟต์แวร์ประยุกต์สามารถจำแนกได้เป็น 2 ประเภท คือ</p> <p>1) ซอฟต์แวร์สำหรับงานเฉพาะด้าน คือ โปรแกรมซึ่งเขียนขึ้นเพื่อการทำงาน เฉพาะอย่างหรือบางครั้งเรียกว่า User's Program เช่น โปรแกรมการทำบัญชีจ่ายเงินเดือน โปรแกรมระบบเช่าซื้อ โปรแกรมการทำสินค้าคงคลัง เป็นต้น ซึ่งแต่ละโปรแกรมจะมีเงื่อนไขหรือแบบฟอร์มแตกต่างกันออกไปตามความต้องการและกฎเกณฑ์ของแต่ละหน่วยงานที่ใช้ซึ่งสามารถดัดแปลงแก้ไขเพิ่มเติม (Modifications) บางส่วนของโปรแกรมได้ เพื่อให้ตรงกับความต้องการของผู้ใช้ และซอฟต์แวร์ประยุกต์ที่เขียนขึ้นนี้โดยส่วนใหญ่ใช้ภาษาระดับสูงเป็นตัวพัฒนา</p> <p>2) ซอฟต์แวร์สำหรับงานทั่วไป เป็นโปรแกรมประยุกต์ที่มีผู้จัดทำไว้ เพื่อใช้ในการทำงานประเภทต่าง ๆ ทั่วไป โดยผู้ใช้งานคนอื่นๆ สามารถนำโปรแกรมนี้ไปประยุกต์ใช้กับข้อมูลของตนได้ แต่จะไม่สามารถดัดแปลงหรือแก้ไขโปรแกรมได้ ผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมเอง ซึ่งเป็นการประหยัดเวลา แรงงาน และค่าใช้จ่ายในการเขียนโปรแกรม นอกจากนี้ยังไม่ต้องใช้เวลามากในการฝึกและปฏิบัติ ซึ่งโปรแกรมสำเร็จรูปนี้จะมีการใช้งานในหน่วยงานที่ขาดบุคลากรที่มีความชำนาญเป็นพิเศษในการเขียนโปรแกรม ดังนั้น การใช้โปรแกรมสำเร็จรูปจึงเป็นสิ่งที่อำนวยความสะดวกและเป็นประโยชน์อย่างยิ่ง ตัวอย่างโปรแกรม สำเร็จรูปที่นิยมใช้ ได้แก่ MSOffice, Lotus, Adobe Photoshop, SPSS, Internet Explorer และเกมต่างๆ</p> <p><b>3. โปรแกรมคอมพิวเตอร์</b></p> <p>โปรแกรมคอมพิวเตอร์ (Computer Program) คือ กลุ่มชุดคำสั่งที่ใช้อธิบายชิ้นงานหรือ กลุ่มงานที่จะประมวลผลโดยคอมพิวเตอร์โปรแกรมคอมพิวเตอร์อาจเรียกว่า ซอฟต์แวร์ แอปพลิเคชันหรือโปรแกรมคอมพิวเตอร์ส่วนใหญ่นั้นเป็นชุดคำสั่งที่ออกแบบตามขั้นตอนวิธีโดยปกติแล้วจะเขียนโดยโปรแกรมเมอร์ หรือสร้างโดยโปรแกรมอื่น ในโปรแกรมคอมพิวเตอร์ชุดหนึ่งๆ อาจเขียนขึ้นด้วยระบบรหัสที่เรียกว่า “ภาษาเครื่อง” ซึ่งจะมีความยากและเหมาะสมสำหรับช่างเทคนิคเฉพาะทาง ภายหลังจึงได้มีการสร้างภาษาโปรแกรมที่ใกล้เคียงภาษามนุษย์มากขึ้น เช่น ภาษาแอสเซมบลี (Assembly) ภาษาซี ( C ) ภาษาโคบอล (COBOL) ภาษาเบสิก (BASIC) ภาษา C# ภาษาจาวา เป็นต้น</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย ความปลอดภัยในการปฏิบัติงานไฟฟ้าและอิเล็กทรอนิกส์เบื้องต้น	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 1</b>
	ชื่อหน่วย องค์ประกอบของคอมพิวเตอร์และโปรแกรม ภาษาซี	สอนครั้งที่ 1-2
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>-</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>		

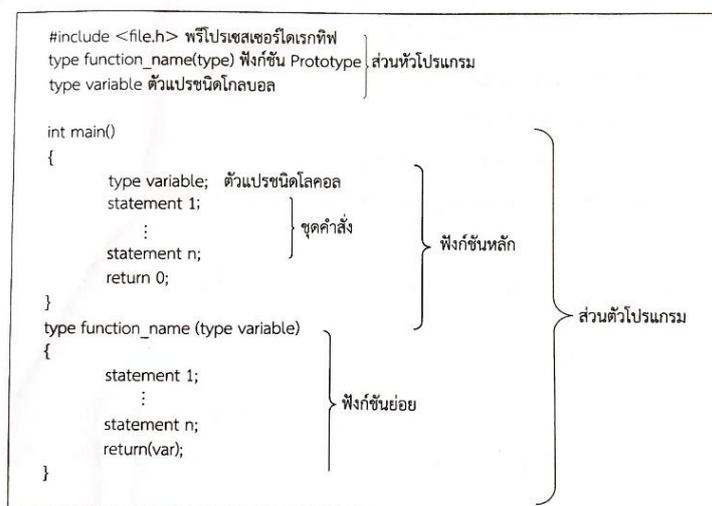
	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและค ภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>1. สาระสำคัญ</b></p> <p>ศึกษาหลักการและขั้นตอนการพัฒนาโปรแกรมคอมพิวเตอร์ เครื่องมือการออกแบบโปรแกรม ความ เป็นมาของภาษาคอมพิวเตอร์ โครงสร้างภาษาคอมพิวเตอร์ ลักษณะเด่นและด้อยของ ภาษาคอมพิวเตอร์ ประเภทข้อมูล องค์ประกอบของคำสั่ง การคำนวณและเปรียบเทียบ ขั้นตอนในการ ทำงานของโอเปอเรชันคำนวณ คำสั่งต่างๆ ในภาษาคอมพิวเตอร์ ฟังก์ชันเบื้องต้น การเขียนโปรแกรมงาน ต่างๆ โดยใช้คำสั่งและฟังก์ชันเบื้องต้นในภาษาใดภาษาหนึ่งกับเครื่องไมโครคอมพิวเตอร์ ปฏิบัติการ วิเคราะห์โจทย์ปัญหา ออกแบบโปรแกรมและเขียนโปรแกรมด้วยคำสั่งภาษาคอมพิวเตอร์ เพื่อให้มีความรู้ ความเข้าใจ และทักษะในการวิเคราะห์โจทย์ปัญหา สามารถออกแบบโปรแกรมและเขียน โปรแกรมใช้งาน รวมทั้งสามารถนำไปประยุกต์ใช้งานได้</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรม ภาษาอื่น ๆ</p> <p>2.2. เขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 โปรแกรม Editor สำหรับเขียนโปรแกรมและคอมไพล์ภาษาซี</p> <p>3.1.2 นักเรียนสามารถปฏิบัติการสืบค้นข้อมูล และสรุปเนื้อหาบทเรียน เรื่อง แนะนำภาษาซี ได้อย่าง ถูกต้อง</p> <p>3.1.3 นักเรียนมีความตั้งใจ มีความรับผิดชอบ ให้ความร่วมมือในการทำงานที่ได้รับมอบหมาย และมี ทัศนคติที่ดีต่อการเรียนรู้เรื่องการเขียนโปรแกรมด้วยภาษาซี</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีวินัย</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 อยู่อย่างพอเพียง</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	หน่วยที่	2
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอ ภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. โครงสร้างของโปรแกรมภาษาซี

โปรแกรมภาษาซีจะประกอบด้วยฟังก์ชันส่วนหัวโปรแกรมที่ใช้เก็บไลบรารี (Library) เพื่อใช้ คอมไพล์ โปรแกรมโดยมีส่วนขยาย คือ .h ส่วนตัวโปรแกรมต้องมีฟังก์ชันชื่อว่า main() อย่างน้อยหนึ่ง ฟังก์ชัน จึงจะ สามารถ Execute Program ได้ ลักษณะของฟังก์ชัน main() จะต้องเป็นฟังก์ชันที่ไม่มีการ ส่งค่าไปยังฟังก์ชัน อื่น หรือไม่มี Argument โดยสามารถใช้คำว่า int นำหน้าฟังก์ชัน ซึ่ง main() สามารถ เขียนได้เป็น int main (void) โดยขอบเขตฟังก์ชัน main (Delimiters) ในโปรแกรมภาษาซีใช้เครื่องหมาย { แทนการเริ่มต้นฟังก์ชัน และใช้เครื่องหมาย } แทนการสิ้นสุดฟังก์ชัน ดังนั้นเมื่อเขียนฟังก์ชัน main() ทุกครั้งจะต้องมีเครื่องหมาย { และ } อยู่ด้วยเสมอ และการปิดท้ายคำสั่งในภาษาซีจะต้องใช้เครื่องหมาย ; (Semicolon) เป็นการบ่งชี้ให้ c Compiler ทราบว่าจบคำสั่ง (statement) แต่ละคำสั่ง ชื่อตัวแปร (Variable Name) สามารถตั้งชื่อโดยใช้ ตัวอักษรตัวเล็กหรือตัวอักษรตัวใหญ่ก็ได้ รูปแบบองค์ประกอบโครงสร้างโปรแกรมภาษาซี มีดังนี้



##### 1.1 ส่วนหัวโปรแกรม ประกอบด้วย

1.1.1 프리โปรเซสเซอร์ไดเรกทีฟ คือ คำสั่งรูปแบบหนึ่งในภาษาซีที่มีความพิเศษ โดยใน ขั้นตอน การแปลความหมายโปรแกรม ถ้าตัวแปลภาษาซีตรวจพบว่ามีการใช้ฟรีโปรเซสเซอร์ไดเรกทีฟภายใน โปรแกรม ฟรีโปรเซสเซอร์ไดเรกทีฟเหล่านั้นจะถูกแปลความหมายเป็นลำดับแรกก่อนคำสั่งประเภทอื่นๆ รูปแบบของการ เขียนฟรีโปรเซสเซอร์ไดเรกทีฟจะต้องขึ้นต้นด้วยเครื่องหมาย # แต่ไม่ต้องลงท้ายด้วย เครื่องหมาย ; เหมือน คำสั่งอื่นโดยทั่วไป คำสั่งที่จัดอยู่ในกลุ่มฟรีโปรเซสเซอร์ไดเรกทีฟแสดงได้ดังนี้

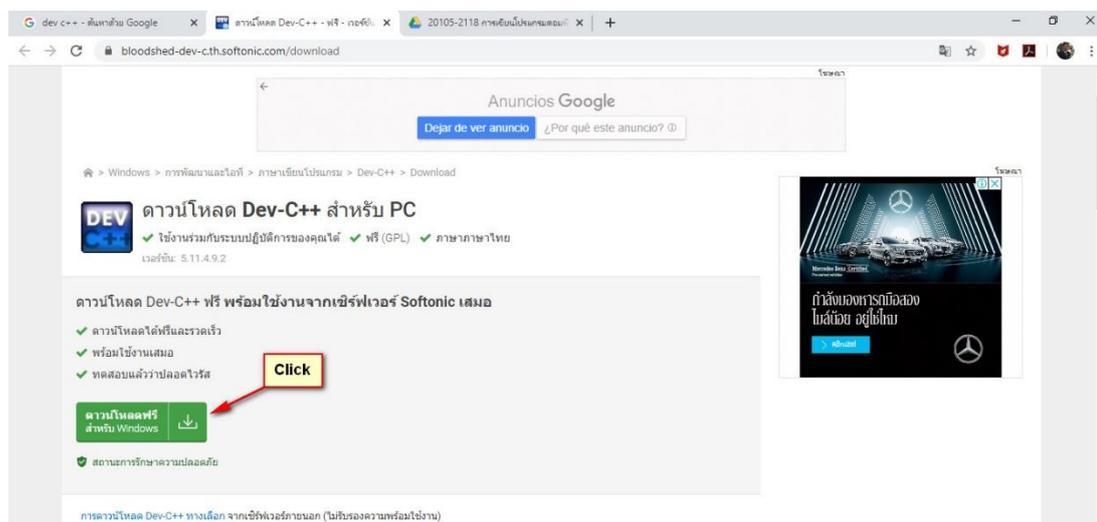
	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอ ภาษาซี	สอนครั้งที่ 3-4
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p>1) <code>#include</code> จะใช้สำหรับสั่งให้ตัวแปลภาษาซีนำไฟล์ที่กำหนดชื่อไว้ต่อจาก <code>include</code> เข้ามารวมกับโปรแกรมก่อนที่จะแปลโปรแกรม เนื่องจากในบางกรณีที่มีการเรียกใช้คำสั่งจากไฟล์อื่นๆ ดังนั้นจึงต้องเขียน <code>#include</code> ไว้ที่ส่วนหัวโปรแกรมเสมอ โดยมีรูปแบบการเขียน 2 แบบ ดังนี้</p> <p>(1) แบบที่ 1 <code>include &lt;ชื่อไฟล์&gt;</code> การใช้เครื่องหมาย <code>&lt;-&gt;</code> ระบุชื่อไฟล์เพื่อให้ตัวแปลภาษาซี เริ่มค้นหาไฟล์จากไดเรกทอรีที่กำหนดไว้ เช่น <code>include &lt;stdio.h&gt;</code> เป็นต้น</p> <p>(2) แบบที่ 2 <code>#include "ชื่อไฟล์"</code> การใช้เครื่องหมายระบุชื่อไฟล์ ตัวภาษาซีจะเริ่มค้นหาไฟล์จากไดเรกทอรีปัจจุบันก่อน ถ้าไม่พบจะไปค้นหาต่อในไดเรกทอรีที่กำหนดไว้ เช่น <code>#include "stdio.h"</code> และ <code>#include "math.h"</code> เป็นต้น</p> <p>2) <code>define</code> แมโคร (Macro) เป็นชื่อที่สร้างขึ้นมาภายในโปรแกรมพร้อมกับกำหนดค่าหรือความหมายให้กับแมโครนั้น โดยมีข้อกำหนดว่าชื่อแมโครต้องพิมพ์เป็นอักษรตัวพิมพ์ใหญ่ และ ค่าที่กำหนดให้กับแมโครสามารถเป็นได้ตั้งแต่ตัวเลข ข้อความ หรือคำสั่งที่ได้ผลลัพธ์ออกมาแน่นอน <code>#define</code> ใช้สำหรับสร้างแมโครและกำหนดค่าให้กับแมโครนั้น รูปแบบการเขียนพรีโพรเซสเซอร์ไดเรกทีฟ <code>#define</code> แสดงได้ดังนี้</p> <pre>#define ชื่อของแมโคร "ค่าที่กำหนดให้แมโคร" #define MESSAGE "Hello" #define FIRST 1000 #define SECOND FIRST+200</pre> <p>1.1.2 ฟังก์ชัน <b>Prototype</b> คือ <code>include</code> ไฟล์เป็นเท็กซ์ไฟล์ธรรมดาที่สามารถเปิดอ่านได้ จะเก็บข้อมูลที่จำเป็นต้องใช้ในการเรียกใช้ฟังก์ชันเช่น การประกาศตัวแปร การประกาศ Prototype ของ ฟังก์ชัน เป็นต้น</p> <p>1.1.3 <b>ตัวแปรโกลบอล</b> คือ ตัวแปรที่สามารถเรียกใช้ที่ตำแหน่งใดๆ ในโปรแกรมก็ได้ ยกเว้น ในกรณีที่มีการประกาศตัวแปรที่มีชื่อเดียวกัน ตัวแปรโกลบอลภายในบล็อกหรือฟังก์ชันซึ่งในตอนนี้ ไม่จำเป็นต้องมีก็ได้</p> <p>1.1 ส่วนตัวโปรแกรม จะประกอบด้วยส่วนของโปรแกรมหลักและส่วนของโปรแกรมน้อยซึ่ง อาจจะไม่มีก็ได้</p> <p>1.2.1 <b>ฟังก์ชันหลัก</b> ประกอบด้วย ฟังก์ชัน <code>main()</code> คือ ฟังก์ชันพื้นฐานของทุกโปรแกรมภาษาซี ซึ่งเหมือน</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอ ภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>ฟังก์ชันอื่นๆ ที่มี Parameter และ Value เพียงแต่ค่าของ Value จะมีการคืนไปที่ระบบปฏิบัติการ แต่ฟังก์ชันอื่นๆ จะคืนมาที่ main() รูปแบบของ int main() จะมีการ return ค่า แต่ในรูปแบบ void main() จะไม่มี การ return ค่ากลับ ภาษาซีจะเริ่มต้นการทำงานที่ฟังก์ชันที่ชื่อ main วิธีการเขียนฟังก์ชัน main สามารถเขียนได้หลายวิธี ได้แก่</p> <pre style="text-align: center;">void main(void) int main(void) void main (int argc, char **argv); int main(int argc, char **argv);</pre> <p>void หรือ int ที่อยู่หน้า main เป็นตัวกำหนดการคืนค่าหลังจากโปรแกรมทำงานจบ ถ้าเป็น int โปรแกรมต้องมีการคืนค่าด้วยคำสั่ง return หรือ exit เช่น return 0; หรือ exit(); เป็นต้น</p> <p>void หรือ int argc, char **argv ในวงเล็บหลัง main เป็นตัวกำหนดว่าโปรแกรมมีParameters หรือไม่</p> <p>1) <b>ตัวแปรโลคอล</b> คือ ขอบเขตการทำงานของตัวแปรแต่ละตัวจะกำหนดอยู่ภายในบล็อกของคำสั่งภายใต้เครื่องหมายปีกกา { } หรือการประกาศในช่วงของการประกาศฟังก์ชันเรียกตัวแปรเหล่านี้ว่า ตัวแปรโลคอล (Local Variable) ใช้สำหรับประกาศตัวแปรที่จะใช้ในเฉพาะฟังก์ชันของตนเอง ฟังก์ชันอื่นไม่สามารถเข้าถึงหรือใช้ได้ ซึ่งจะต้องประกาศตัวแปรก่อนการใช้งานเสมอและจะต้องประกาศ ไว้ในส่วนนี้เท่านั้น</p> <p>2) <b>ชุดคำสั่ง</b> คือ คำสั่งควบคุมการทำงานของโปรแกรม ซึ่งแบ่งกลุ่มลักษณะการทำงานตามข้อกำหนดมาตรฐานของสถาบัน ANSI (American National Standards Institute) กำหนดให้ภาษาซี มีคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม</p> <p><b>ฟังก์ชันย่อย</b> คือ ฟังก์ชันเพื่อนิยามการทำงานที่จะเรียกใช้ในส่วนต่างๆ ของโปรแกรม โดยฟังก์ชันการทำงานดังกล่าวจะถูกเขียนไว้ในฟังก์ชันเพียงครั้งเดียวเท่านั้น แต่สามารถเรียกใช้งาน ได้หลายครั้ง ตัวแปรที่ประกาศหรือคำสั่งที่เรียกในฟังก์ชันใดๆ มีขอบเขตการใช้งานอยู่ในฟังก์ชันนั้นๆ หมายความว่าฟังก์ชันอื่นๆ รวมทั้งฟังก์ชัน main() จะไม่ทราบการทำงานภายในหรือคำสั่งต่างๆในฟังก์ชันนั้น หรือ กล่าวได้ว่าฟังก์ชันแต่ละฟังก์ชันจะไม่ทราบการทำงานภายในของฟังก์ชันอื่น แต่จะสามารถเรียกใช้งาน ฟังก์ชันนั้นได้</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอม ภาษาซี	<b>สอนครั้งที่</b>	3-4
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	8

## 2. วิธีการดาวน์โหลดโปรแกรม Dev-C++

Dev-C++ เป็นโปรแกรมโอเพนซอร์ส (Open Source) สามารถดาวน์โหลดและใช้งานได้ฟรี ใช้สำหรับเขียนภาษา C, C++ โปรแกรม Dev-C++ เป็นโปรแกรม IDE (Integrated Development Environment) หรือก็คือเครื่องมือที่ช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมมี รายละเอียดโปรแกรม Dev-C++ สามารถดาวน์โหลดโปรแกรม Dev-C++ ได้ที่ <https://bloodshed-dev-c.th.softonic.com/download>



รูปที่ 2.1 เว็บไซต์สำหรับดาวน์โหลดโปรแกรม Dev-C++

## 3. คุณสมบัติของ Dev-C++

- Support GCC-based compilers
- Integrated debugging (using GDB)
- Project Manager
- Customizable syntax highlighting editor
- Class Browser
- Code Completion
- Function listing

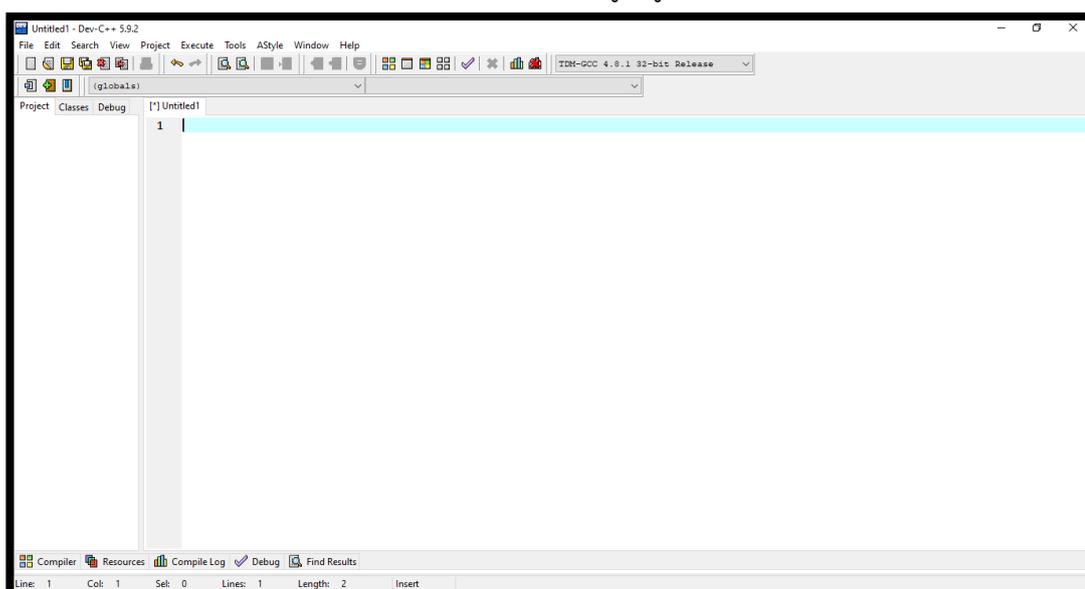
	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและค ภาษาซี	<b>สอนครั้งที่</b>	<b>3-4</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>8</b>

- Profiling support
- Quickly create Windows, console, static libraries and DLLs
- Support of templates for creating your own project types
- Make file creation
- Edit and compile Resource files
- Tool Manager
- Print support
- Find and replace facilities
- CVS support

#### 4. วิธีการเขียนภาษาซีด้วยโปรแกรม Dev-C++

Dev-C++ เป็น Editor ที่ใช้ในการเขียนโปรแกรมด้วยภาษาซี และ C++ ตัวโปรแกรมมีขนาดเล็กใช้  
ง่าย

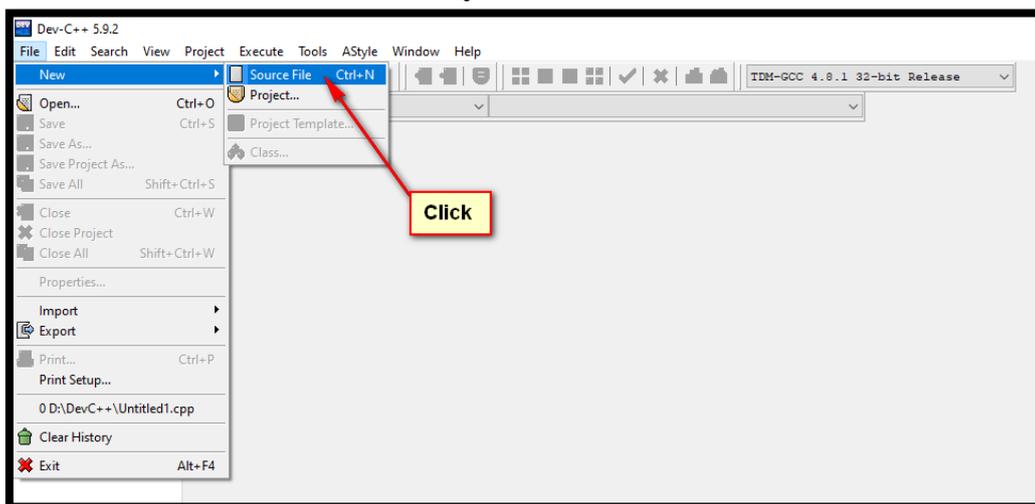
ในหนังสือเล่มนี้จะเป็นการฝึกเขียนโปรแกรมภาษาซีผ่านโปรแกรม Dev-C++ เนื่องจากเป็นโปรแกรมที่ติดตั้ง  
และใช้งานง่าย หลังจากติดตั้งโปรแกรม Dev-C++ ลงในเครื่องแล้ว เรียกโปรแกรม Dev-C++ ขึ้นมาจะปรากฏ  
หน้าต่างของโปรแกรมประกอบด้วย Editor และคอมไพเลอร์อยู่ดังรูปที่ 2.2



รูปที่ 2.2 หน้าต่างของโปรแกรม

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	2
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

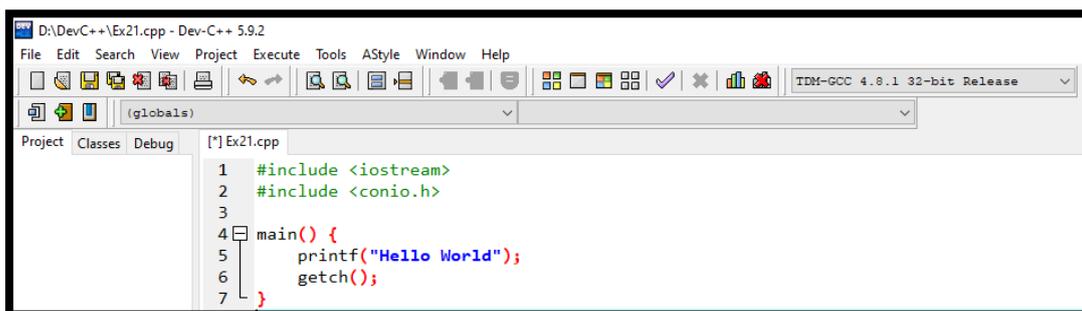
การเขียนภาษาซีด้วยโปรแกรม Dev-C++ ไปที่เมนู File -> New -> Source File เพื่อการเปิดไฟล์ใหม่



รูปที่ 2.3 การเปิดไฟล์และเลือกประเภทไฟล์

## 5. การป้อนคำสั่งภาษาซีลงในโปรแกรม Dev-C++

การป้อนคำสั่งภาษาซีลงในโปรแกรม Dev-C++ สามารถทำได้โดยการพิมพ์คำสั่งต่างๆ ของ ภาษาซีลงไปในหน้าต่าง Editor Screen ดังรูปที่ 2.4 ทั้งนี้คำสั่งที่พิมพ์ลงไปจะต้องถูกหลังไวยากรณ์ของคำสั่งของภาษาซีที่ได้มีกำหนดเอาไว้



รูปที่ 2.4 การป้อนคำสั่งภาษาซีลงในโปรแกรม

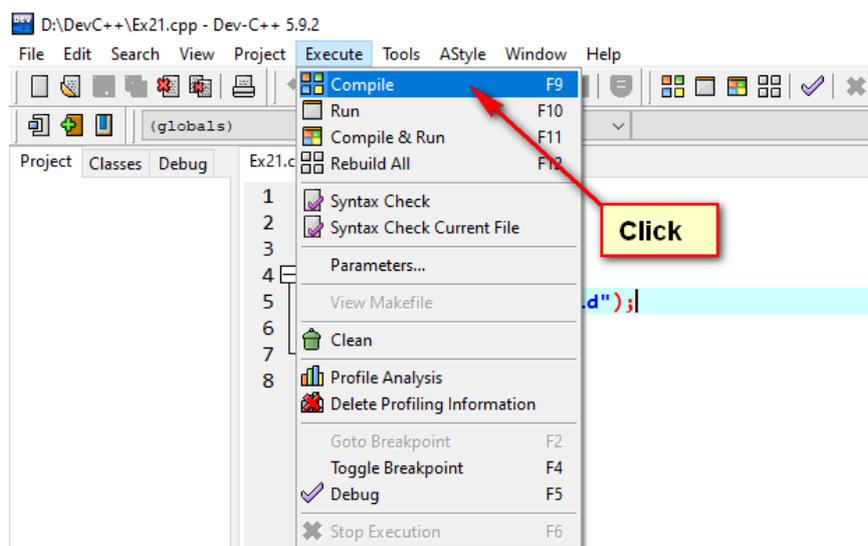
## 6. การจัดเก็บโปรแกรมที่เขียนขึ้นจากภาษาซีด้วยโปรแกรม Dev-C++

ในการจัดเก็บโปรแกรมที่เขียนขึ้น ผู้ใช้งานสามารถนำกลับมาใช้งานอีกครั้งในภายหลัง ซึ่งกระบวนการจัดเก็บจะมีความคล้ายกับการจัดเก็บเอกสารต่างๆ โดยจัดเก็บโปรแกรมที่พัฒนาขึ้นจากโปรแกรม Dev-C++ สามารถทำได้โดยการไปที่เมนู File -> Save หรือกด ctrl+s

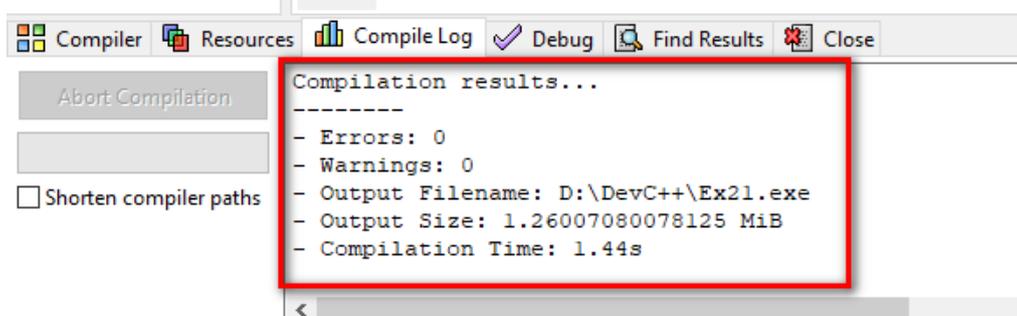
	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	2
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

### ขั้นตอน 7. การแปลภาษาซีหรือการคอมไพล์โปรแกรม

การแปลภาษาซีหรือการคอมไพล์ (Compile) คือ การตรวจสอบโปรแกรมที่เขียนขึ้นถูกต้อง ตามหลักไวยากรณ์ของคำสั่งของภาษาซีหรือไม่ หรือกล่าวอีกนัยหนึ่งว่าการคอมไพล์ในภาษาซี คือ การแปลโปรแกรมภาษาซีให้เป็นภาษาเครื่อง หากแปลได้หมดก็สามารถสั่งให้โปรแกรมทำงานได้ แต่หากแปล ไม่ได้ก็จะแจ้งข้อผิดพลาดให้ทราบผู้เรียกคอมไพล์ก็ต้องแก้ไขโปรแกรมภาษาซีให้ถูกต้องเสียก่อนจึงเรียก คอมไพล์ใหม่โดยการแปลภาษาซีหรือการคอมไพล์ (Compile) จากโปรแกรม Dev-C++ สามารถทำได้ โดยการไปที่ใช้เมนู Execute -> Compile ดังรูปที่ 2.5 หรือกดปุ่มที่ F9 ที่คีย์บอร์ด เมื่อมีการคอมไพล์จะมีการแสดงสถานะ การคอมไพล์ (Compile) คือ การตรวจสอบโปรแกรมที่เขียนขึ้นถูกต้องตามหลักไวยากรณ์ของคำสั่งของ ภาษาซีดังรูปที่ 2.6



รูปที่ 2.5 การคอมไพล์

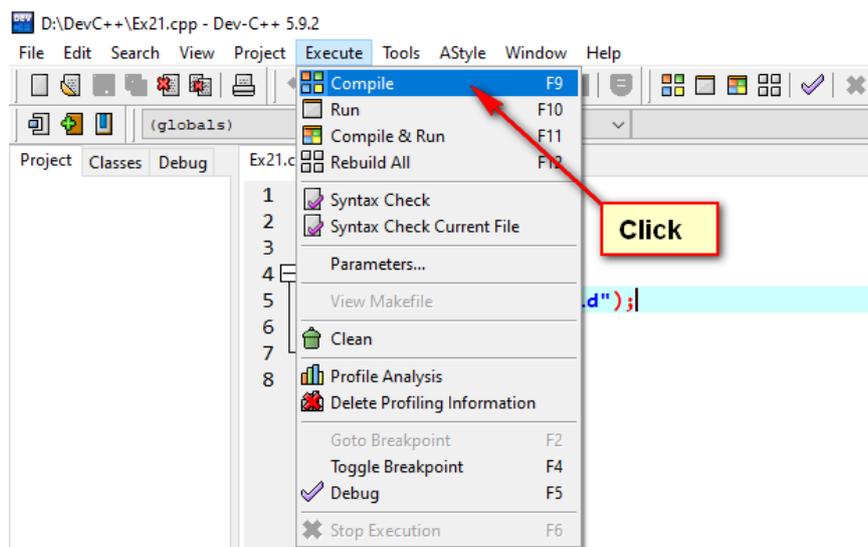


รูปที่ 2.6 การแสดงสถานะของโปรแกรมที่เกิดขึ้นหลังจากคอมไพล์

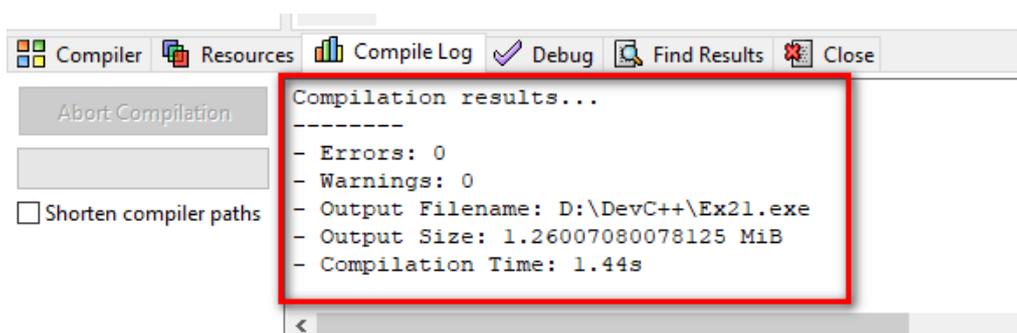
	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	2
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอ ภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

### ขั้นตอน 7. การแปลภาษาซีหรือการคอมไพล์โปรแกรม

การแปลภาษาซีหรือการคอมไพล์ (Compile) คือ การตรวจสอบโปรแกรมที่เขียนขึ้นถูกต้อง ตามหลักไวยากรณ์ของคำสั่งของภาษาซีหรือไม่ หรือกล่าวอีกนัยหนึ่งว่าการคอมไพล์ในภาษาซี คือ การแปลโปรแกรมภาษาซีให้เป็นภาษาเครื่อง หากแปลได้หมดก็สามารถสั่งให้โปรแกรมทำงานได้ แต่หากแปล ไม่ได้ก็จะแจ้งข้อผิดพลาดให้ทราบผู้เรียกคอมไพล์ก็ต้องแก้ไขโปรแกรมภาษาซีให้ถูกต้องเสียก่อนจึงเรียก คอมไพล์ใหม่โดยการแปลภาษาซีหรือการคอมไพล์ (Compile) จากโปรแกรม Dev-C++ สามารถทำได้ โดยการไปที่ใช้เมนู Execute -> Compile ดังรูปที่ 2.5 หรือกดปุ่มที่ F9 ที่คีย์บอร์ด เมื่อมีการคอมไพล์จะมีการแสดงสถานะ การคอมไพล์ (Compile) คือ การตรวจสอบโปรแกรมที่เขียนขึ้นถูกต้องตามหลักไวยากรณ์ของคำสั่งของ ภาษาซีดังรูปที่ 2.6



รูปที่ 2.5 การคอมไพล์

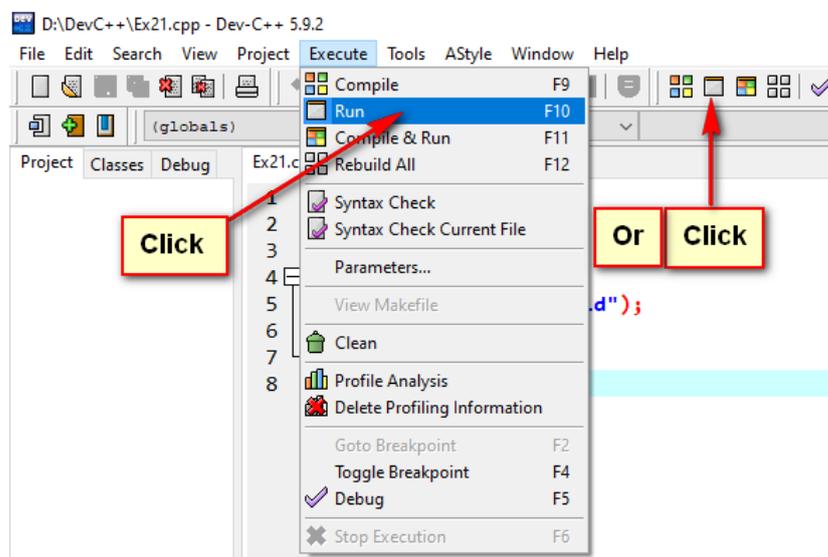


รูปที่ 2.6 การแสดงสถานะของโปรแกรมที่เกิดขึ้นหลังจากคอมไพล์

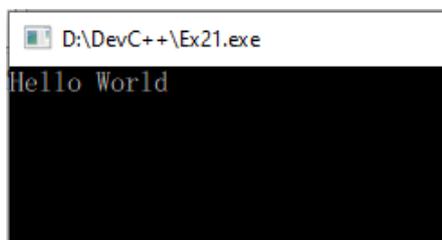
	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	2
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอภาษาซี	สอนครั้งที่	3-4
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

#### 8. การสั่งให้โปรแกรมทำงานหรือการทดสอบการทำงานของโปรแกรม

เมื่อทำการแก้ไขในส่วนที่ผิดพลาดเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการสั่งให้โปรแกรมทำงานหรือทดสอบการทำงานของโปรแกรมที่เรียกว่าการรันโปรแกรม (Run) การรันโปรแกรมจากโปรแกรม Dev-C++ ทำได้โดยเลือกเมนู Execute -> Run หรือคลิกที่ Tool bar  หรือกด F10 ที่คีย์บอร์ด ดังรูปที่ 2.7 หลังจากสั่งให้โปรแกรมทำงานแล้วผลลัพธ์ของโปรแกรมจะปรากฏในจอภาพอีกจอภาพหนึ่ง โดยจอภาพดังกล่าวแสดงข้อความ ดังรูปที่ 2.8



รูปที่ 2.5 การคอมไพล์



รูปที่ 2.8 ผลลัพธ์จากการรันโปรแกรม

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและค ภาษาซี	<b>สอนครั้งที่</b>	<b>3-4</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>8</b>
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>2</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและค ภาษาซี	<b>สอนครั้งที่</b>	<b>3-4</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>8</b>
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน์ (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>1</b>
	ชื่อหน่วย โปรแกรม Editor สำหรับเขียน โปรแกรมและคอ ภาษาซี	<b>สอนครั้งที่</b>	<b>3-4</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>8</b>
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>-</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 3</b>
	ชื่อหน่วย การเขียนโปรแกรมเบื้องต้นด้วยผังงาน	สอนครั้งที่ 5-6
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>1. สาระสำคัญ</b></p> <p>ผังงาน(Flow Chart) คือรูปภาพหรือสัญลักษณ์ ที่ใช้เขียนแทนคำอธิบาย ข้อความหรือคำพูดที่ใช้ในอัลกอริธึม เพราะการเข้าใจขั้นตอนที่ง่ายและตรงกันนั้น การใช้คำพูดหรือข้อความอาจทำได้ยากกว่าการใช้รูปภาพหรือสัญลักษณ์</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. นักเรียนสามารถปฏิบัติการเขียนผังงาน Flowchart ได้อย่างถูกต้อง</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 นักเรียนสามารถอธิบายความรู้เกี่ยวกับผังงาน Flowchart ได้อย่างถูกต้อง</p> <p>3.1.2 นักเรียนสามารถปฏิบัติการเขียนผังงาน Flowchart ได้อย่างถูกต้อง</p> <p>3.1.3 นักเรียนมีความตั้งใจ มีความรับผิดชอบ ให้ความร่วมมือในการทำงานที่ได้รับมอบหมาย และมีทัศนคติที่ดีต่อการเรียนรู้เรื่องการเขียนโปรแกรมด้วยภาษาซี</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 ใฝ่เรียนรู้</p> <p>3.3.2 มีเหตุผล</p> <p>3.3.3 รู้จักกาลเทศะ</p>		

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 3
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่ 5-6
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ผังงาน

ผังงาน (Flowchart) คือ ผังที่ใช้รูปภาพมาแสดงขั้นตอนการทำงานของโปรแกรมโดยจะใช้สัญลักษณ์ต่างๆ เพื่อใช้แทนรูปแบบหรือการทำงานของคำสั่งโปรแกรม

##### 1.1 ประโยชน์ของผังงาน มีดังนี้

- 1.1.1 ทำให้เข้าใจและแยกแยะปัญหาได้ง่าย (Problem Define)
- 1.1.2 แสดงลำดับการทำงาน (Step Flowing)
- 1.1.3 หาข้อผิดพลาดได้ง่าย (Easy to Debug)
- 1.1.4 ทำความเข้าใจโปรแกรมได้ง่าย (Easy to Read)
- 1.1.5 ไม่ขึ้นกับภาษาใดภาษาหนึ่ง (Flexible Language)

##### 1.2 หลักการเขียนผังงานที่ดี มีดังนี้

- 1.2.1 ใช้สัญลักษณ์ตามที่กำหนดไว้
- 1.2.2 ใช้ลูกศรแสดงทิศทางการไหลของข้อมูลจากบนลงล่างหรือจากซ้ายไปขวา
- 1.2.3 คำอธิบายในภาพควรสั้นกะทัดรัดและเข้าใจง่าย
- 1.2.4 ทุกแผนภาพต้องมีลูกศรแสดงทิศทางเข้า-ออก
- 1.2.5 ไม่ควรโยงเส้นเชื่อมผังงานที่อยู่ใกล้กันมากๆ ควรใช้สัญลักษณ์จุดเชื่อมต่อแทน
- 1.2.6 ผังงานควรมีการทดสอบความถูกต้องของการทำงานก่อนนำไปเขียนโปรแกรม

##### 1.3 หลักการจัดทิศทางของภาพผังงาน มีดังนี้

- 1.3.1 ทิศทางเริ่มจากบนลงล่างและจากซ้ายไปขวา
- 1.3.2 สัญลักษณ์ที่ใช้ต้องเป็นตามมาตรฐาน อาจมีขนาดต่างๆ กัน แต่รูปร่างเป็นสัดส่วนตาม

มาตรฐาน

1.3.3 หลีกเลี่ยงการโยงเส้นไปมาในทิศทางที่ตัดกันให้ใช้เครื่องหมายแสดงจุดต่อเนื่องได้ และ ทุกจุดในผังงานไม่มีการปล่อยจุดใดจุดหนึ่งไว้ลอยๆ

- 1.3.4 ใช้คำอธิบายที่กะทัดรัดและเข้าใจได้ง่าย

1.3.5 มีความเป็นระเบียบเรียบร้อย สะอาด และควรเขียนชื่อของผังงาน ชื่อผู้เขียน วันที่และเลขลำดับหน้าเสมอ

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 3</b>
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่ 5-6
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

ดังนั้น ผังงาน (Flowchart) เป็นเครื่องมือหนึ่งที่ใช้อธิบายลำดับขั้นตอนการทำงานในรูปแบบแผนภาพ โดยใช้สัญลักษณ์รูปร่างต่างๆ ที่มีความหมายแทนคำสั่งและใช้ข้อความในสัญลักษณ์แทนข้อมูลตัวแปร ดำเนินการทางการคำนวณ และการเปรียบเทียบ สัญลักษณ์ของผังงาน (Flowchart) แสดงดังตาราง

#### ตารางแสดงสัญลักษณ์ที่ใช้ในการผังงาน

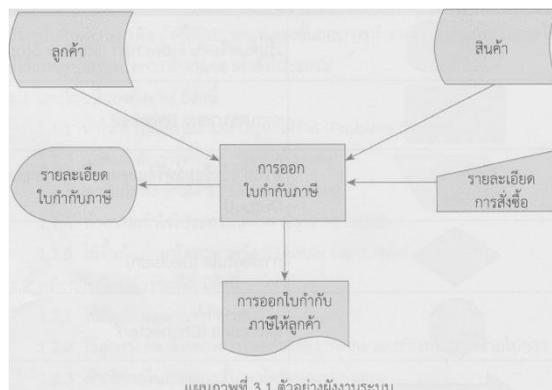
รูปภาพสัญลักษณ์	ความหมายของสัญลักษณ์
	เริ่มต้นหรือจบ Flowchart (Start or Stop)
	การประมวลผล (Process)
	ส่วนนำเข้าข้อมูลหรือแสดงผลข้อมูล (Input or Output)
	การตัดสินใจ (Decision)
	จุดเชื่อมต่อ (Connector)
	ทิศทางการทำงาน (Direction of Flow)

## 2. ประเภทของผังงาน

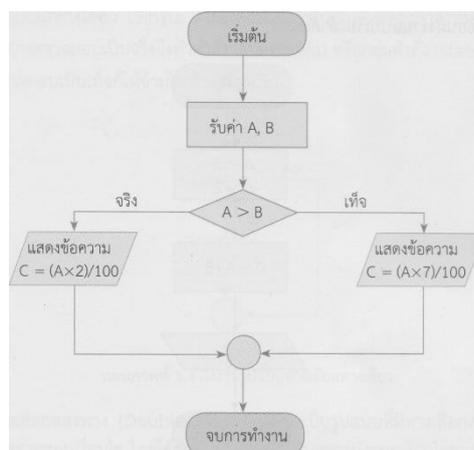
การเขียนผังงานสามารถแบ่งลักษณะการทำงานและความสัมพันธ์เป็นรูปแบบต่างๆ ได้แก่ การทำงานแบบมีลำดับ การทำงานแบบมีเงื่อนไข และการทำงานแบบทำซ้ำภายใต้เงื่อนไขต่างๆ หลังจากนั้น จึงนำผังงาน (Flowchart) ที่ออกแบบไว้ไปเขียนเป็นภาษาคอมพิวเตอร์ ประเภทของผังงานโดยทั่วไป •ปงเป็น 2 ประเภทใหญ่ คือ

2.1 ผังงานระบบ (System Flowchart) เป็นผังงานที่แสดงถึงขั้นตอนการทำงานภายในระบบ\* หนึ่งๆ เพื่อให้เห็นโครงสร้างโดยภาพรวมของระบบ ซึ่งจะแสดงถึงความเกี่ยวข้องของส่วนที่สำคัญต่างๆ ในระบบนั้น เช่น เอกสารข้อมูลเบื้องต้น สื่อบันทึกข้อมูลที่ใช้ ข้อมูลจะส่งผ่านไปยังหน่วยงานใด มีกิจกรรมประมวลผล ข้อมูลอะไรในหน่วยงานนั้น แล้วจะส่งต่อไปหน่วยงานใด ซึ่งแต่ละจุดจะประกอบไปด้วยการนำเข้าข้อมูล (Input) วิธีการประมวลผล (Process) และการแสดงผลลัพธ์ (Output) ดังแผนภาพที่ 3.1 แสดงความสัมพันธ์ระหว่างงานต่างๆ ในระบบว่ามีความสัมพันธ์กัน หน่วยงานที่เกี่ยวข้องในการออา ใบก้ากับภาษีตั้งแต่เรื่องสินค้า ลูกค้า และการออกใบกำกับภาษี

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 3</b>
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่ 5-6
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8



2.2 ผังงานโปรแกรม (Program Flowchart) ผังงานประเภทนี้แสดงถึงขั้นตอนของ คำสั่งที่ใช้ในโปรแกรม ผังงานนี้อาจสร้างจากผังงานระบบโดยเป็นภาพแผนผังที่แสดงลำดับขั้นตอน ในการทำงานของโปรแกรม ซึ่งจะแยกย่อยมาจากผังงานระบบ ใบแต่ละขั้นตอนจะแสดงการทำงาน"น แต่ละคำสั่งโดยละเอียด ใส่วิธีการ และจัดลำดับขั้นตอนของโปรแกรมสำหรับโปรแกรมนั้นๆ ตั้งแต่ เริ่มต้นจากการรับข้อมูล การประมวลผล และไปจนถึงการแสดงผลลัพธ์ที่ได้จากการประมวลผลV ผู้เขียนโปรแกรมกำลังทำงานอยู่ไม่ได้แสดงความสัมพันธ์ระหว่างโปรแกรมนั้นกับโปรแกรมอื่นๆ ดังแผนภาพที่ 3.2 ซึ่งแสดงการรับค่าข้อมูลแบบจำนวนเต็มเข้ามาเก็บในตัวแปร A และ B แล้ว เปรียบเทียบค่าในตัวแปร A กับ B ถ้า  $A > B$  จะแสดงค่าผลลัพธ์ที่ได้ในตัวแปร  $C = (A \times 2) / 100$  แต่ถ้า  $A = B$  แสดงค่าผลลัพธ์ในตัวแปร  $C = (A \times 7) / 100$

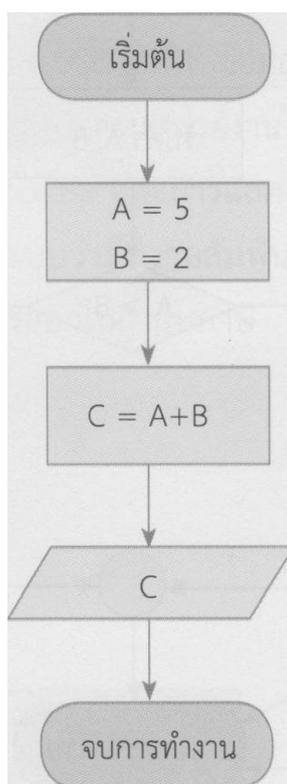


	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	3
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

### 3. การเขียนผังงานการทำงานแบบเรียงลำดับ

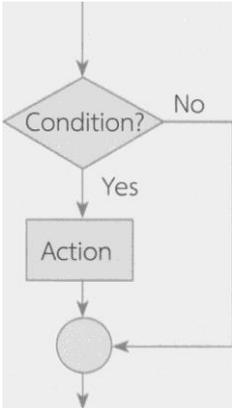
ขั้นตอนการทำงานแบบเรียงลำดับเป็นขั้นตอนวิธีการทำงานพื้นฐาน การเขียนอธิบายการทำงานวิธีการแก้ปัญหาโดยใช้ผังงานทำให้สามารถเห็นลำดับขั้นตอนวิธีการทำงานของการแก้ปัญหาเป็นภาษา -ตอรีโดยตรงสำหรับผู้เริ่มต้นฝึกหัดการเขียนโปรแกรม การเขียนอธิบายขั้นตอนวิธีการทำงานในลักษณะของผังงานแบบเรียงลำดับในตอนแรกคือการใช้ผังงานในการออกแบบโปรแกรมโดยเริ่มจากผังงานจำนวนของกระบวนการมายังผังงานโปรแกรมเป็นเทคนิคการออกแบบจากบนลงล่าง (Top-down Design) ในการแยกจำนวนของกระบวนการทำงานทั้งหมดของผังงาน ให้พิจารณาว่าผังงานนั้นมีการทำทั้งหมดที่กระบวนการในการทำงานของกระบวนการหนึ่งนั้นจะมีทิศทางเข้าออกได้อย่างละหนึ่งทิศทางเท่านั้น และอธิบายขั้นตอนได้จำนวนหนึ่งข้อความเท่านั้น ดังตัวอย่างที่ 1

#### ตัวอย่างที่ 1 การเขียนผังงานแบบเรียงลำดับ



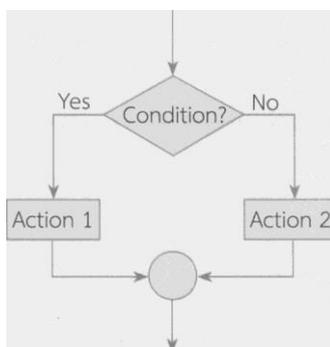
แผนภาพที่ 3.3 ผังงานการบวกเลข 2 จำนวน

อธิบายผังงานการบวกเลข 2 จำนวน

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>3</b>
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>อธิบายผังงานการบวกเลข 2 จำนวน</p> <ol style="list-style-type: none"> <li>1) เริ่มต้นการทำงาน</li> <li>2) กำหนดค่าให้ตัวแปร โดยให้ <math>A = 5</math> และ <math>B = 2</math></li> <li>3) กำหนดตัวแปร C เก็บค่าของผลบวก จะได้ <math>c = A+B</math></li> <li>4) แสดงค่าที่เก็บไว้ในตัวแปร C</li> <li>5) จบการทำงาน</li> </ol> <p>4. การเขียนผังงานการทำงานแบบมีเงื่อนไข</p> <p>การเขียนขั้นตอนวิธีการทำงานแบบมีเงื่อนไขให้ใช้สำหรับกรณีที่ต้องการเลือกวิธีการทำงานสำหรับการแก้ไข ปัญหาที่เหมาะสมกับลักษณะข้อมูลในขณะนั้น รูปแบบของการเขียนผังงานจะมีการตัดสินใจ โดยขึ้นกับเงื่อนไข (Condition) ในการตัดสินใจว่ามีค่าความเป็นจริงหรือเท็จ และโปรแกรมก็จะข้ามไปทำงานในกลุ่มของคำสั่ง ตามเงื่อนไขนั้นๆ รูปแบบของคำสั่งการตัดสินใจโดยทั่วไป มีดังนี้</p> <p><b>4.1 ทางเลือกทางเดียว (Single Alternative)</b> เป็นลักษณะของคำสั่งที่มีทางเลือกทางเดียว คือ ถ้าเงื่อนไขในการตรวจสอบเป็นจริงจึงทำคำสั่ง (Statement) หรือกลุ่มคำสั่ง (Statement Block) แต่ ถ้าเงื่อนไขการตรวจสอบเป็นเท็จก็ให้ข้ามไปทำคำสั่งถัดไป</p> <div style="text-align: center;">  </div> <p style="text-align: center;">แผนภาพที่ 3.4 ผังงานรูปแบบทางเลือกทางเดียว</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	3
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

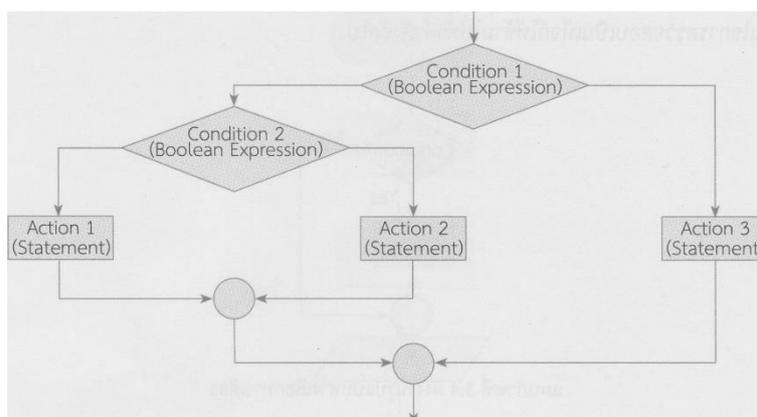
**4.2 ทางเลือกสองทาง (Double Alternative)** เป็นรูปแบบที่มีทางเลือกสองทางเลือกขึ้นอยู่กับค่าที่ใช้ในการตรวจสอบเงื่อนไข โดยใช้คำสั่ง ELSE เพื่อเป็นการแบ่งแยกเงื่อนไขของทางเลือก คือ เมื่อ เงื่อนไขในการตรวจสอบมีค่าความจริงเป็นจริง ก็จะไปทำคำสั่งหลังคำว่า THEN แล้วจึงทำคำสั่งถัดจาก คำสั่ง END IF และถ้าเงื่อนไขการตรวจสอบมีค่าความจริงเป็นเท็จ ก็จะไปทำคำสั่งหลังคำว่า ELSE แล้วจึง ทำคำสั่งถัดจาก คำสั่ง END IF ต่อไป



แผนภาพที่ 3.5 ผังงานรูปแบบทางเลือกสองทาง

**4.3 ทางเลือกหลายทาง (Multiple Alternative)** โครงสร้างควบคุมแบบการตัดสินใจชนิด "งเลือกหลายทางยังสามารถแบ่งย่อยได้อีก 2 ลักษณะ คือ

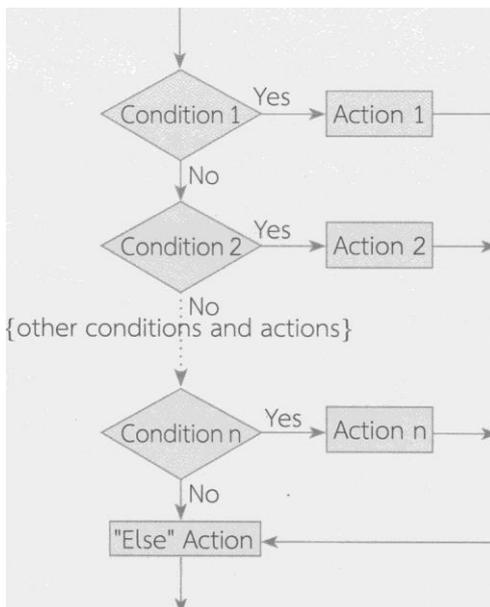
**4.3.1 ทางเลือกหลายทางแบบ IF ซ้อน IF (Nested IF statement)** ดังแผนภาพที่ 3.6



แผนภาพที่ 3.6 ผังงานรูปแบบทางเลือกแบบ Nested IF

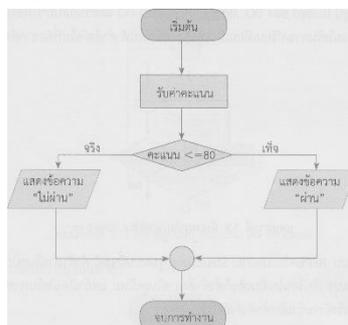
	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	3
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

#### 4.3.2 ทางเลือกหลายทางแบบ Case (Case Control Structure) ดังแผนภาพที่ 3.7

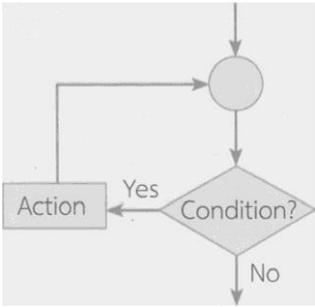


แผนภาพที่ 3.7 ผังงานรูปแบบทางเลือกแบบ Case (Case Control structure)

ตัวอย่างที่ 2 การเขียนผังงานแบบมีเงื่อนไข



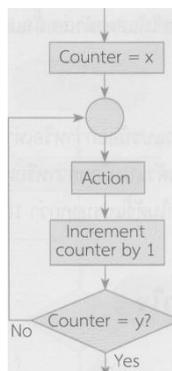
แผนภาพที่ 3.8 การเขียนผังงานแบบมีเงื่อนไข

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>3</b>
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>อธิบายผังงานแบบมีเงื่อนไข</p> <ol style="list-style-type: none"> <li>1) เริ่มต้นการทำงาน</li> <li>2) กำหนดตัวแปรให้รับคะแนน</li> <li>3) ตรวจสอบคะแนนว่าน้อยกว่าหรือเท่ากับ 80 หรือไม่</li> <li>4) ให้แสดงผลว่า “ไม่ผ่าน” เมื่อคะแนนน้อยกว่าหรือเท่ากับ 80</li> <li>5) ให้แสดงผลว่า “ผ่าน” เมื่อคะแนนมากกว่า 80</li> <li>6) จบการทำงาน</li> </ol> <p><b>5. การเขียนผังงานการทำงานแบบทำซ้ำ</b></p> <p>วิธีการเขียนโครงสร้างผังงานแบบทำซ้ำ มีวิธีการเขียนต่างไปจากการทำงานแบบเรียงลำดับและแบบมีเงื่อนไข แต่บางครั้งการทำงานของขั้นตอนวิธีการทำงานแบบทำซ้ำอาจประกอบด้วยขั้นตอนวิธีการทำงานแบบลำดับหรือขั้นตอนวิธีการทำงานแบบมีเงื่อนไข โครงสร้างการทำซ้ำหรือการวนรอบนั้นสามารถใช้ได้ 3 รูปแบบ ได้แก่</p> <p><b>5.1 รูปแบบ WHILE Structure</b> ในการเปรียบเทียบเงื่อนไขของการทำซ้ำ ถ้าเงื่อนไขเป็นจริงก็ จะทำคำสั่งภายในลูป (Loop) ของ DO...WHILE จนถึง END DO และวนกลับมาเปรียบเทียบอีกครั้งเพื่อ ทำงานต่อไป จนเงื่อนไขในการเปรียบเทียบเป็นเท็จจึงจะข้ามไปทำคำสั่งถัดไปที่ต่อจากคำสั่ง END DO</p> <div style="text-align: center;">  </div> <p style="text-align: center;">แผนภาพที่ 3.9 ผังรูปแบบ WHILE Structure</p> <p><b>5.2 รูปแบบ REPEAT...UNTIL Structure</b> รูปแบบนี้จะทำคำสั่งภายในลูปก่อนหนึ่งครั้งแล้วเปรียบเทียบค่าตัวแปร ถ้าเงื่อนไขเป็นเท็จก็ทำคำสั่งภายในลูปใหม่ แต่ถ้าเงื่อนไขในการเปรียบเทียบหลัง คำสั่ง UNTIL เป็นจริงจึงจะข้ามไปทำคำสั่งถัดไป</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	3
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

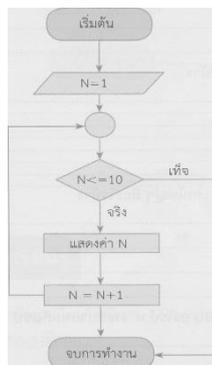
แผนภาพที่ 3.10 ผังงานรูปแบบ REPEAT...UNTIL structure

5.3 รูปแบบ DO...END DO structure (FOR Loop) หากทราบจำนวนที่จะวนรอบที่แน่นอน ควรจะใช้ For Loop โดยมี Loop Index เป็นตัวควบคุมการวนรอบ หากตรวจสอบจำนวนรอบจะกระทำโดยอัตโนมัติ การสิ้นสุดการวนรอบจะกระทำเมื่อเงื่อนไขเป็นเท็จ



แผนภาพที่ 3.11 ผังงานรูปแบบ DO...END DO structure lbs

ตัวอย่างที่ 3 การเขียนผังงานแบบทำซ้ำ



แผนภาพที่ 3.12 ผังงานการทำงานแบบทำซ้ำ

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	3
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่	5-6
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>อธิบายผังงานแผนภาพที่ 3.12 โดยให้แสดงตัวเลขตั้งแต่ 1 ถึง 10 ออกทางจอภาพ</p> <ol style="list-style-type: none"> <li>1) เริ่มต้นการทำงาน</li> <li>2) กำหนดตัวแปร <math>N = 1</math></li> <li>3) ตรวจสอบค่า <math>N</math> ว่าค่าในตัวแปรน้อยกว่าหรือเท่ากับ 10 หรือไม่</li> <li>4) ให้แสดงผลค่า <math>N</math> เมื่อค่าในตัวแปรน้อยกว่าหรือเท่ากับ 10</li> <li>5) จบการทำงานค่า <math>N</math> เมื่อค่าในตัวแปรมากกว่า 10</li> </ol>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 3</b>
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่ 5-6
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 3</b>
	<b>ชื่อหน่วย</b> กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	<b>สอนครั้งที่</b> 5-6
		<b>ชั่วโมงรวม</b> 72
		<b>จำนวนชั่วโมง</b> 8
<b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b>		
<b>6.1 สื่อสิ่งพิมพ์</b>		
6.1.1 ใบความรู้		
6.1.2 หนังสือ		
<b>6.2 สื่อโสตทัศน (ถ้ามี)</b>		
6.2.1 มัลติมีเดีย		
6.2.2 Power Point		
<b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b>		
-		
<b>6.4 อื่น ๆ (ถ้ามี)</b>		
-		
<b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b>		
7.1.1 ใบความรู้		
7.1.2 ใบงาน แบบฝึกหัด		
7.1.3 หนังสือเรียน (Textbook)		
<b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b>		
8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน		
8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 3</b>
	ชื่อหน่วย กฎของโอห์มและวงจรไฟฟ้าเบื้องต้น	สอนครั้งที่ 5-6
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>4</b>
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>1. สาระสำคัญ</b></p> <p>ในการเขียนโปรแกรมต้องมีการใช้งานข้อมูล ข้อมูลต่างๆ เหล่านี้จะมีการนำไปแสดงผลและรับเข้ามาโดยผ่านอุปกรณ์อินพุตและเอาต์พุต เนื่องจากข้อมูลมีหลายรูปแบบตามชนิดที่มีให้ใช้ในภาษาซี ผู้เขียนโปรแกรมจึงต้องทราบวิธีการที่จะนำมาแสดงผลและรับค่าข้อมูลได้ถูกต้องตรงตามความต้องการ</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับการแสดงผลข้อมูลและการรับค่า</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 เข้าใจหลักการเขียนโปรแกรมด้วยคอมไพเตอร์</p> <p>3.1.2 มีทักษะในการวางแผน เขียน ตรวจสอบ และแก้ไขโปรแกรมคอมไพเตอร์</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีวินัย</p> <p>3.3.2 ตรงต่อเวลา</p> <p>3.3.3 ใฝ่เรียนรู้</p>			



	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	4
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

ตารางที่ 4.1 คำสงวน (Reserved Word)

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

## 2. ชนิดของข้อมูล

ชนิดของข้อมูล (Data Type) ในการเขียนโปรแกรมต่างๆ จะมีข้อมูลต่างๆ เข้ามาเกี่ยวข้อง เช่น การนับจำนวนวนรอบ (Loop) ของการทำงานโดยใช้ข้อมูลชนิดจำนวนเต็ม หรือการแสดงความยาวโดยใช้ข้อมูลชนิดอักขระ จะเห็นว่าข้อมูลต่างๆ ถูกแบ่งออกเป็นหลายชนิดตามจุดประสงค์ของการใช้งาน นอกจากนี้ข้อมูลแต่ละชนิดยังใช้เนื้อที่หน่วยความจำ (Memory) ไม่เท่ากัน

ตารางที่ 4.2 ชนิดของข้อมูล (Data Type)

ชนิดของตัวแปร	ขนาด (Bits)	ขอบเขต	ข้อมูลที่เก็บ
char	8	-128 ถึง 127	ข้อมูลชนิดอักขระ ใช้เนื้อที่ 1 byte
unsigned char	8	0 ถึง 255	ข้อมูลชนิดอักขระ ไม่คิดเครื่องหมาย
int	16	-32,768 ถึง 32,767	ข้อมูลชนิดจำนวนเต็ม ใช้เนื้อที่ 2 byte
unsigned int	16	0 ถึง 65,535	ข้อมูลชนิดจำนวนเต็ม ไม่คิดเครื่องหมาย
short	8	-128 ถึง 127	ข้อมูลชนิดจำนวนเต็มแบบสั้น ใช้เนื้อที่ 1 byte
unsigned snort	8	0 ถึง 255	ข้อมูลชนิดจำนวนเต็มแบบสั้น ไม่คิดเครื่องหมาย
long	32	-2,147,483,648 ถึง 2,147,483,647	ข้อมูลชนิดจำนวนเต็มแบบยาว ใช้เนื้อที่ 4 byte

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	4
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

#### ตารางที่ 4.2 ชนิดของข้อมูล (Data Type) (ต่อ)

ชนิดของตัวแปร	ขนาด (Bits)	ขอบเขต	ข้อมูลที่เก็บ
unsigned long	32	0 ถึง 4,294,967,295	ข้อมูลชนิดจำนวนเต็มแบบยาว ไม่คิดเครื่องหมาย
float	32	$1.2 \times 10e(-38)$ ถึง $3.4 \times 10e(38)$	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 4 byte
double	64	$2.3 \times 10e(-308)$ ถึง $1.7 \times 10e(308)$	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 8 byte

### 3. ค่าคงที่ในภาษาซี

ค่าคงที่ (Constants) จะต่างจากตัวแปรตรงที่ค่าคงที่จะเก็บค่าเพียงค่าเดียวตลอดทั้งโปรแกรม โดยเมื่อสร้างค่าคงที่แล้วจะไม่สามารถเปลี่ยนแปลงค่าได้ในการตั้งชื่อค่าคงที่จะใช้กฎเดียวกันกับการตั้งชื่อ ตัวแปร แต่นิยมตั้งชื่อค่าคงที่ให้เป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด เพื่อให้เกิดความแตกต่างระหว่างชื่อตัวแปร กับชื่อค่าคงที่ ค่าคงที่ในภาษาซีมี 2 คำสั่ง คือ

**3.1 Define** สามารถกำหนดชื่อค่าคงที่ซึ่งใช้อยู่ประจำและไม่มีการเปลี่ยนแปลงค่า โดยการใช้ `#define` ด้วยรูปแบบดังนี้

```
#define identifier value
```

ตัวอย่าง การประกาศค่าคงที่โดยใช้คำสั่ง `#define` มีดังนี้

```
#define PI 3.14159265
```

```
#define NEWLINE '\n'
```

```
#define WIDTH 100
```

**3.2 Constants** สามารถใช้กำหนดค่าให้กับตัวแปรที่ไม่ต้องการจะเปลี่ยน โดยคำคำสั่ง `const` มีข้อดี คือ

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>4</b>
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>3.2.1 ในการสร้างชื่อสัญลักษณ์สามารถกำหนดชนิดข้อมูลของค่าคงที่ได้ตามต้องการ(คำสั่ง #define ไม่สามารถทำได้)</p> <p>3.2.2 ใช้กฎของกรอบ (Scoping Rule) เพื่อจำกัดอาณาบริเวณของชื่อสัญลักษณ์ให้อยู่แต่เฉพาะในฟังก์ชันใดฟังก์ชันหนึ่งหรือในไฟล์ใดไฟล์หนึ่งเท่านั้น</p> <p>3.2.3 ใช้ const กับชนิดข้อมูลสืบทอดอย่างเช่นอาร์เรย์ได้โดยจะยกตัวอย่างการใช้งาน const ดังนี้</p> <div style="border: 1px solid black; padding: 5px; text-align: center; margin: 10px auto; width: fit-content;"> <pre>const datatype variable name = value;</pre> </div> <p>ตัวอย่างการประกาศค่าคงที่โดยใช้คำสั่ง const มีดังนี้</p> <pre>const float a = 5.23; const int b = a%2;</pre> <p><b>4. ฟังก์ชันการแสดงผล</b></p> <p>การแสดงผลลัพธ์ คือ การแสดงออกทางจอภาพในภาษาซีในรูปแบบของฟังก์ชันโดยฟังก์ชันมาตรฐานที่มีในไลบรารีเพื่อแสดงผลทางจอภาพ เช่น printf(), puts(), putchar() เป็นต้น ฟังก์ชันเหล่านี้อยู่ใน Header File ชื่อ stdio.h</p> <p><b>4.1 ฟังก์ชัน printf()</b> เป็นฟังก์ชันที่แสดงผลออกทางจอภาพ สามารถกำหนดรูปแบบและรหัสควบคุมได้ โดยฟังก์ชัน printf() มีรูปแบบ ดังนี้</p> <div style="border: 1px solid black; padding: 5px; text-align: center; margin: 10px auto; width: fit-content;"> <pre>Printf (control string,argument list);</pre> </div> <p>control string คือ รหัสรูปแบบข้อมูลที่ใช้ในการพิมพ์ผลลัพธ์ฟังก์ชัน printf() สามารถใช้รหัสควบคุมเพื่อช่วยจัดรูปแบบการพิมพ์ผลลัพธ์ให้สวยงามขึ้น โดยที่ control string จะต้องเขียนอยู่ภายใต้เครื่องหมาย “ ” (Double Quotation)</p> <p>argument list คือค่าคงที่ ตัวแปร หรือนิพจน์ในกรณีที่มีค่าคงที่ ตัวแปร หรือค่านิพจน์หลายๆ ค่า ให้ใช้เครื่องหมาย , (Comma) คั่นระหว่างค่าคงที่ ตัวแปร หรือนิพจน์แต่ละค่า</p> <p>ตัวอย่าง การแสดงข้อความอย่างเดียว printf(“I love C language”)</p> <p>ความหมาย แสดงข้อความ I love C language</p> <p>ตัวอย่าง การแสดงค่าตัวแปรอย่างเดียว printf(“%d”,num);</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>4</b>
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>ความหมาย แสดงค่าในตัวแปรชื่อ num</p> <p><b>4.2 ฟังก์ชัน putchar()</b> เป็นฟังก์ชันที่แสดงผลออกจากจอภาพทีละ 1 ตัวอักขระ โดยลักษณะของฟังก์ชันที่เป็นฟังก์ชันที่ต้องการ argument 1 ค่าที่เป็นข้อมูลชนิด Single Character (char) รูปแบบการใช้งานฟังก์ชัน มีดังนี้</p> <div style="text-align: center; border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <pre>putchar (char_argument);</pre> </div> <p>putchar() คือ ฟังก์ชันที่ใช้ <span style="float: right;">แสดงผลลัพธ์ทีละ 1 ตัวอักขระ</span> ออกทางจอภาพ</p> <p>char_argument คือ ตัวแปรชนิด Single Character (char)</p> <p><b>4.3 ฟังก์ชัน puts()</b> เป็นฟังก์ชันที่พิมพ์ข้อความแสดงออกทางจอภาพโดยลักษณะของฟังก์ชันนี้เป็นฟังก์ชันที่ต้องการ Argument 1 ค่าที่เนนชนิดข้อความ (string Constant) รูปแบบการใช้งานฟังก์ชัน มีดังนี้</p> <div style="text-align: center; border: 1px solid gray; padding: 5px; width: fit-content; margin: 10px auto;"> <pre>puts(string_argument);</pre> </div> <p>Puts() คือ ฟังก์ชันที่ใช้พิมพ์ข้อความออกทางจอภาพ</p> <p>string_argument คือ ค่าคงที่ชนิดสตริง (string Constant) ซึ่งค่าคงที่สตริงนี้จะถูกพิมพ์ออกทางจอภาพผ่านฟังก์ชัน puts()</p> <p><b>5. รหัสควบคุม</b></p> <p>หากต้องการการแสดงผลที่เนนระเบียบจะต้องใช้รหัสควบคุมการแสดงผล ซึ่งสามารถจัดรูปแบบการแสดงผลให้ดูเนนระเบียบได้ เช่น ขึ้นบรรทัดใหม่หลังแสดงข้อความหรือเว้นระยะแต่้ระหว่างข้อความโดยใช้รหัสควบคุมการแสดงผลร่วมกับคำสั่ง printf() ในภาษาซี มีรหัสควบคุมแสดงผลหลายรูปแบบด้วยกัน ดังนี้</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	4
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

ตารางที่ 4.3 รหัสควบคุมข้อมูลที่สามารถใช้ในฟังก์ชัน printf()

รหัสรูปแบบ (Format Code)	ความหมาย	ค่า ASCII
\a	เสียงดังกอกลำโพงหนึ่งครั้ง	0x07
\b	เลื่อน Cursor ไปลบตัวอักษรทางซ้ายมือหนึ่ง	0x08
\f	ขึ้นหน้าใหม่	0x0c
\n	ขึ้นบรรทัดใหม่	0x0a
\r	เลื่อน Cursor ไปทางซ้ายมือสุดของบรรทัด	0x0d
\t	ตั้ง Tab ในแนวนอน	0x09
\v	ตั้ง Tab ในแนวตั้ง	0x0b
\\	เครื่องหมาย\	0x5c
\'	เครื่องหมาย'	0x2c
\"	เครื่องหมาย"	0x22
\?	เครื่องหมาย ?	0x3f

ตารางที่ 4.4 แสดงรหัสรูปแบบข้อมูลที่สามารถใช้ในฟังก์ชัน printf(), scanf()

รหัสรูปแบบ (Format Code)	ความหมาย
%c	ใช้กับข้อมูลชนิดตัวอักษรตัวเดียว (char: Single Character)
%d	ใช้กับข้อมูลชนิดตัวเลขจำนวนเต็ม (int: Integer) โดยสามารถใช้กับตัวเลขฐาน 10 เท่านั้น
%e	ใช้กับข้อมูลชนิดตัวเลขจุดทศนิยม (float: Floating Point)
%f, %lf	ใช้กับข้อมูลชนิด float และ double ตามลำดับ
%g	ใช้กับข้อมูลชนิด float
%h	ใช้กับข้อมูลชนิด Short Integer
%l	ใช้กับข้อมูลชนิด int โดยใช้กับตัวเลขฐาน 8, ฐาน 10 และฐาน 16

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	4
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
%o	ใช้กับข้อมูลชนิด int โดยสามารถใช้กับตัวเลขฐาน 8 เท่านั้น		
%u	ใช้กับข้อมูลชนิด unsigned int โดยใช้กับตัวเลขฐาน 10 เท่านั้น		
%x	ใช้กับข้อมูลชนิด int โดยสามารถใช้กับตัวเลขฐาน 16 เท่านั้น		
%s	ใช้แสดงข้อมูลชนิด String		
%p	ใช้แสดงค่า Address ของตัวแปร'พอยน์เตอร์'		

**6. ฟังก์ชันการรับค่าข้อมูล**

ภาษาซีมีฟังก์ชันที่ใช้ในการรับข้อมูลจากคีย์บอร์ด คือ ฟังก์ชัน scanf(), ฟังก์ชัน getchar(), ฟังก์ชัน getch(), ฟังก์ชัน getche() และฟังก์ชัน gets() ซึ่งแต่ละฟังก์ชันมีรายละเอียดของการทำงานดังนี้

**6.1 ฟังก์ชัน scanf()** เป็นฟังก์ชันที่ใช้รับข้อมูลจากคีย์บอร์ดเข้าไปเก็บไว้ในตัวแปรที่กำหนดไว้ โดยสามารถรับข้อมูลที่เป็นตัวเลขจำนวนเต็ม ตัวเลขทศนิยม ตัวอักขระตัวเดียว หรือข้อความก็ได้ รูปแบบ การใช้งานฟังก์ชัน มีดังนี้

```
scanf (control string, argument list);
```

control string คือ รหัสรูปแบบข้อมูล (Format Code) โดยจะต้องเขียนอยู่ภายใต้เครื่องหมาย "" (Double Quotation) และห้ามมีข้อความใดๆ ในส่วน Control

argument list คือ ส่วนของ Argument List เป็นตัวแปรที่จะรับค่าจากแป้นพิมพ์มาเก็บไว้ ถ้า มีมากกว่าหนึ่งค่าจะต้องแยกกันด้วยเครื่องหมายคอมม่า (,) และเนื่องจากเป็นการรับข้อมูลมาเก็บในหน่วยความจำแทนการรับค่าจากหน่วยความจำไปแสดง ดังนั้นการใช้ scanf จึงต้องมีอักขระ & นำหน้า ตัวแปรเสมอ ยกเว้นการรับข้อความ (String) จะไม่ต้องใช้เครื่องหมายนี้

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>4</b>
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	<b>สอนครั้งที่</b>	7-8
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	8
<p><b>6.2 ฟังก์ชัน getchar()</b> เป็นฟังก์ชันที่รับข้อมูลจากคีย์บอร์ดเพียง 1 ตัวอักษรโดยการรับข้อมูล ของฟังก์ชันนี้จะต้องกดแป้น Enter ทุกครั้งที่ป้อนข้อมูลเสร็จ จึงทำให้เห็นข้อมูลที่ป้อนปรากฏบนจอภาพ ด้วย ถ้าต้องการนำข้อมูลที่ป้อนผ่านทางคีย์บอร์ดไปใช้งาน จะต้องกำหนดตัวแปรชนิด Single Character (char) ขึ้นมา 1 ตัว เพื่อเก็บค่าข้อมูลที่รับผ่านทางคีย์บอร์ด ในทางตรงกันข้ามถ้าไม่ต้องการใช้ข้อมูลที่ป้อน ผ่านทางคีย์บอร์ดก็ไม่จำเป็นต้องกำหนดตัวแปรชนิด char ขึ้นมา รูปแบบการใช้งานฟังก์ชัน มีดังนี้</p> <p style="text-align: center;">Getchar(); หรือ char_var = getchar();</p> <p>Getchar() คือ ฟังก์ชันที่รับข้อมูลเพียง 1 ตัวอักษรจากคีย์บอร์ดโดยฟังก์ชันนี้จะไม่ Argument ซึ่งอาจจะใช้ getchar(void) แทนคำว่า getchar() ก็ได้ แต่นิยมใช้ getchar()</p> <p>char var คือ ตัวแปรชนิด char ซึ่งจะเก็บข้อมูล 1 ตัวอักษรที่ป้อนผ่านทางคีย์บอร์ด</p> <p><b>6.3 ฟังก์ชัน getch()</b> เป็นฟังก์ชันที่รับข้อมูลเพียง 1 ตัวอักษรเหมือนกับฟังก์ชัน getchar() แตกต่างกันตรงที่เมื่อใช้ฟังก์ชันนี้รับข้อมูล ข้อมูลที่ป้อนเข้าไปจะไม่ปรากฏให้เห็นบนจอภาพและไม่ต้องกด Enter ตามรูปแบบการใช้งานฟังก์ชัน มีดังนี้</p> <p style="text-align: center;">getch(); หรือ char_var = getch();</p> <p>getch() คือ ฟังก์ชันที่ได้รับข้อมูลเพียง 1 ตัวอักษรจากคีย์บอร์ดโดยฟังก์ชันนี้จะไม่ Argument ดังนั้นอาจจะใช้ getch(void) แทนคำว่า getch() ก็ได้ แต่นิยมใช้ getch()</p> <p>char_var คือ ตัวแปรชนิด char ซึ่งจะเก็บข้อมูล 1 ตัวอักษรที่ป้อนผ่านทางคีย์บอร์ด</p> <p><b>6.4 ฟังก์ชัน getche()</b> เป็นฟังก์ชันที่รับข้อมูลจากคีย์บอร์ดเพียง 1 ตัวอักษรเหมือนฟังก์ชัน แตกต่างกันตรงที่ข้อมูลที่ป้อนเข้าไป จะปรากฏให้เห็นบนจอภาพด้วย ส่วนมีการทำงานและลักษณะ ใช้งาน อื่นๆ เหมือนฟังก์ชัน getch() ทุกประการ รูปแบบการใช้งานฟังก์ชัน มีดังนี้</p> <p style="text-align: center;">getche(); หรือ char_var = getche();</p> <p>getche() คือ ฟังก์ชันที่รับข้อมูลเพียง 1 ตัวอักษรจากคีย์บอร์ด โดยฟังก์ชันนี้จะไม่ Argument ดังนั้นอาจจะใช้ getche(void) แทนคำว่า getche() ก็ได้ แต่นิยมใช้ getche()</p> <p>char_var คือ ตัวแปรชนิด char ซึ่งจะเก็บข้อมูล 1 ตัวอักษรที่ป้อนผ่านทางคีย์บอร์ด</p> <p><b>6.5 ฟังก์ชัน gets()</b> เป็นฟังก์ชันที่รับข้อมูลชนิดข้อความ (String) จากคีย์บอร์ด จากนั้นนำ ข้อมูลที่รับเข้าไปเก็บไว้ในตัวแปรสตริง (String Variables) ที่กำหนดไว้ รูปแบบการใช้งานฟังก์ชัน มีดังนี้</p> <p style="text-align: center;">gets(string_var);</p> <p>string_var คือ ตัวแปรสตริง ซึ่งจะเก็บข้อมูลชนิดข้อความ (string Constant)</p> <p>gets() คือ ฟังก์ชันที่รับข้อความจากคีย์บอร์ดแล้วไปเก็บไว้ในตัวแปรสตริง</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>4</b>
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p>3.2.1 ในการสร้างชื่อสัญลักษณ์สามารถกำหนดชนิดข้อมูลของค่าคงที่ได้ตามต้องการ(คำสั่ง #define ไม่สามารถทำได้)</p> <p>3.2.2 ใช้กฎของกรอบ (Scoping Rule) เพื่อจำกัดอาณาบริเวณของชื่อสัญลักษณ์ให้อยู่แต่เฉพาะในฟังก์ชันใดฟังก์ชันหนึ่งหรือในไฟล์ใดไฟล์หนึ่งเท่านั้น</p> <p>3.2.3 ใช้ const กับชนิดข้อมูลสืบทอดอย่างเช่นอาร์เรย์ได้โดยจะยกตัวอย่างการใช้งาน const ดังนี้</p> <div style="border: 1px solid black; padding: 5px; text-align: center; margin: 10px auto; width: fit-content;"> <pre>const datatype variable name = value;</pre> </div> <p>ตัวอย่างการประกาศค่าคงที่โดยใช้คำสั่ง const มีดังนี้</p> <pre>const float a = 5.23;</pre> <pre>const int b = a%2;</pre> <p><b>4. ฟังก์ชันการแสดงผล</b></p> <p>การแสดงผลลัพธ์ คือ การแสดงออกทางจอภาพในภาษาซีในรูปแบบของฟังก์ชันโดยฟังก์ชันมาตรฐานที่มีในไลบรารีเพื่อแสดงผลทางจอภาพ เช่น printf(), puts(), putchar() เป็นต้น ฟังก์ชันเหล่านี้อยู่ใน Header File ชื่อ stdio.h</p> <p><b>4.1 ฟังก์ชัน printf()</b> เป็นฟังก์ชันที่แสดงผลออกทางจอภาพ สามารถกำหนดรูปแบบและรหัสควบคุมได้ โดยฟังก์ชัน printf() มีรูปแบบ ดังนี้</p> <div style="border: 1px solid black; padding: 5px; text-align: center; margin: 10px auto; width: fit-content;"> <pre>Printf (control string,argument list);</pre> </div> <p>control string คือ รหัสรูปแบบข้อมูลที่ใช้ในการพิมพ์ผลลัพธ์ฟังก์ชัน printf() สามารถใช้รหัสควบคุมเพื่อช่วยจัดรูปแบบการพิมพ์ผลลัพธ์ให้สวยงามขึ้น โดยที่ control string จะต้องเขียนอยู่ภายใต้เครื่องหมาย “ ” (Double Quotation)</p> <p>argument list คือค่าคงที่ ตัวแปร หรือนิพจน์ในกรณีที่มีค่าคงที่ ตัวแปร หรือค่านิพจน์หลายๆ ค่า ให้ใช้เครื่องหมาย , (Comma) คั่นระหว่างค่าคงที่ ตัวแปร หรือนิพจน์แต่ละค่า</p> <p>ตัวอย่าง การแสดงข้อความอย่างเดียว printf(“I love C language”)</p> <p>ความหมาย แสดงข้อความ I love C language</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	4
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	4
	ชื่อหน่วย การแสดงผลข้อมูลและการรับค่า	สอนครั้งที่	7-8
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่ 9-10
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p><b>1. สาระสำคัญ</b></p> <p>พจน์ (Expressions) ทางคณิตศาสตร์นั้นเป็นนิพจน์ที่ใช้ในการคำนวณ ซึ่งนิพจน์ทางคณิตศาสตร์นั้นจะมีรูปแบบเหมือนกับสมการคณิตศาสตร์ แต่จะประกอบไปด้วย ค่าคงที่หรือตัวแปร ซึ่งเรียกอีกอย่างว่า ตัวถูกดำเนินการ (Operand) แล้วเชื่อมกันด้วยเครื่องหมายทางคณิตศาสตร์ หรือเรียกอีกอย่างว่า ตัวดำเนินการ (Operator) นั้นเอง</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรมภาษาอื่น ๆ</p> <p>2.2. เขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 เข้าใจหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p>3.1.2 มีทักษะในการวางแผน เขียน ตรวจสอบ และแก้ไขโปรแกรมคอมพิวเตอร์</p> <p>3.1.3 มีเจตคติที่ดีต่อวิชาชีพ มีกิจนิสัยในการค้นคว้าเพิ่มเติม ปฏิบัติงานด้วยความละเอียดรอบคอบ คำนึงถึงความถูกต้องปลอดภัย</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีวินัย</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 อยู่อย่างพอเพียง</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่ 9-10
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ตัวดำเนินการกำหนดค่า

ตัวดำเนินการ หมายถึง ตัวที่ทำหน้าที่รวมค่าหรือกระทำกับค่าต่างๆ ของข้อมูลและตัวแปรตามคำสั่ง และตัวดำเนินการกำหนดค่าเป็นตัวดำเนินการพื้นฐานที่ใช้ในการกำหนดค่าต่างๆ โดยใช้เครื่องหมายเท่ากับ (=) มีรูปแบบของนิพจน์กำหนดค่า คือ

**ตัวแปร = นิพจน์;**

โดยที่นิพจน์อาจจะเป็นค่าคงที่ตัวแปรหรือนิพจน์ที่ประกอบขึ้นจากตัวดำเนินการต่างๆ ก็ได้ เช่น

age = 10;

speed = distance / time;

total\_min = 60 \* hours + minutes;

ch = 'A';

นอกจากนี้ยังมีรูปแบบเพิ่มเติมหากต้องการกำหนดค่าให้กับตัวแปรหลายตัวด้วยค่าเดียวกัน c รูปแบบดังนี้

**ตัวแปร 1 = ตัวแปร 2 = นิพจน์;**

โดยตัวแปรทั้งหมดจะต้องเป็นตัวแปรประเภทเดียวกัน เช่น

int x, y, z;

x = y = z = 45;

แปลความหมายได้ว่า

Z = 45;

y = z; และ x = y;

**ตัวอย่างที่ 1** แสดงการกำหนดค่าจำนวนทศนิยมให้กับตัวแปรจำนวนเต็ม

```

1 #include <stdio.h>
2 void main( )
3 {
4     int x;
5     x = 14.8328;
6     printf("x value is %d", x);
7 }

```

ผลการรันโปรแกรม :

x value is 14

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 5
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่ 9-10
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

## 2. ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์ในภาษาซีประกอบด้วยการบวก ลบ คูณ หาร และหารเอาเศษ การเขียนนิพจน์คณิตศาสตร์ทางคอมพิวเตอร์จะแตกต่างกับนิพจน์ทางคณิตศาสตร์ตรงที่มีลำดับความสำคัญ (Precedence) ของตัวดำเนินการเข้ามาเกี่ยวข้อง โดยตัวดำเนินการคณิตศาสตร์และการทำงานแสดง ดังตารางที่ 5.1

ตารางที่ 5.1 ตัวดำเนินการคณิตศาสตร์และการทำงาน

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
+	บวก (Addition)	$X + y$
-	ลบ (Subtraction)	$X - y$
*	คูณ (Multiplication)	$X * y$
/	หาร (Division)	$X / y$
++	เพิ่มค่าครั้งละ 1 (Increment)	$X++$
--	ลดค่าครั้งละ 1 (Decrement)	$X--$
%	หารเอาผลลัพธ์เฉพาะเศษ (Modulus)	$X \% y$

## ตัวอย่างที่ 2 โปรแกรมหาผลรวมของจำนวนเต็ม 2 จำนวนที่รับข้อมูลจากผู้ใช้

```

1 #include <stdio.h>
2 void main( )
3 {
4     int x, y, z;
5     printf("Enter X value : ");
6     scanf("%d", &x);
7     printf("Enter Y value : ");
8     scanf("%d", &y);
9     z = x + y;
10    printf("Summary of X and Y is %d", z);
11 }
```

ผลการรันโปรแกรม :

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่ 9-10
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

Enter X value : 10  
 Enter Y value : 20  
 Summary of X and Y is 30

### 3. ตัวดำเนินการแบบผสม

ตัวดำเนินการกำหนดค่าแบบผสมเป็นตัวดำเนินการที่ผสมระหว่างตัวดำเนินการกำหนดค่าและตัวดำเนินการทางคณิตศาสตร์ แสดงดังตารางที่ 5.2

ตารางที่ 5.2 ตัวดำเนินการกำหนดค่าแบบผสม

ตัวดำเนินการ	ตัวอย่างคำสั่ง	คำสั่งเต็ม
*=	3*=2;	a = a * 2;
/=	a /= 2;	a = a / 2;
%=	a %= 2;	a = a % 2;
+=	3 += 2;	3 = 3 + 2)
- =	a -= 2;	a = a - 2;

การแปลคำสั่งของตัวดำเนินการกำหนดค่าแบบผสมจะทำคำสั่งที่อยู่ทางขวามือก่อนเสมอ เช่น

$$a *= 3 + 2 ; \text{ จะเท่ากับคำสั่ง } a = a * (3 + 2);$$

### 4. ตัวดำเนินการเพิ่มค่าและลดค่า

ตัวดำเนินการเพิ่มค่าและลดค่าเป็นตัวดำเนินการเพื่อใช้เพิ่มค่าตัวแปรขึ้น 1 หรือลดค่าตัวแปรลง 1 โดยใช้เครื่องหมาย ++ แทนการเพิ่มค่าขึ้น 1 และ -- แทนการลดค่าลง 1 และสามารถในตัวดำเนินการเพิ่มค่าหรือลดค่ากับตัวแปรได้ 2 ตำแหน่ง คือ วางตัวดำเนินการเพิ่มค่าหรือลดค่าไว้หน้าตัวแปรและวางไว้ หลังตัวแปร เช่น

a++; หรือ ++a; ทั้ง 2 คำสั่งจะมีค่าเท่ากับ a = a + 1; และคำสั่ง

a--; หรือ --a; จะมีค่าเท่ากับ a = a - 1;

ความแตกต่างของตำแหน่งการวางตัวดำเนินการเพิ่มค่าหรือลดค่าจะส่งผลถึงการทำงานของ คำสั่งนั้น เช่น

b = 5;

a = 10 + b++ \* 2;

ผลลัพธ์ที่ได้จะมีผลเท่ากับคำสั่ง

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่	9-10
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

$a = 10 + b * 2$ ;  $b = b + 1$ ;

จะได้ว่า  $a$  มีค่าเท่ากับ  $10 + 5 * 2 = 20$  และ  $b$  มีค่าเท่ากับ  $6$

แต่หากนำตัวดำเนินการเพิ่มค่ามาไว้ด้านหน้าตัวแปร โดยกำหนดให้  $b$  มีค่าเท่ากับ  $5$  เช่นเดียวกัน

$b = 5$ ;

$a = 10 + ++b * 2$ ;

ผลลัพธ์ที่ได้จะมีผลเท่ากับคำสั่ง

$b = b + 1$ ;

$a = 10 + b * 2$ ;

จะได้  $b$  มีค่าเท่ากับ  $6$  และ  $a$  มีค่าเท่ากับ  $10 + 6 * 2 = 22$

ตัวอย่างที่ 3 การใช้ตัวดำเนินการเพิ่มค่า

```

1 #include <stdio.h>
2 void main( )
3 {
4     int y, count;
5     count = 1;
6     y = count++;
7     printf("y = %d, count = %d", y, count);
8     count = 1;
9     y = ++count;
10    printf("y = %d, count = %d", y, count);

```

ผลการรันโปรแกรม

$y = 1, count = 2$

$y = 2, count = 2$

$y = 1, count = 2$

$y = 2, count = 2$

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่	9-10
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

### 5. ตัวดำเนินการเปลี่ยนชนิดข้อมูล

การเปลี่ยนชนิดข้อมูลในภาษาซีมี 2 ลักษณะ คือ การเปลี่ยนชนิดข้อมูลโดยไม่ต้องใช้คำสั่ง (Implicit Casting) และการเปลี่ยนชนิดข้อมูลโดยใช้คำสั่ง (Explicit Casting) เช่น หากต้องแปลงข้อมูลชนิด float ไปเป็นข้อมูลชนิด int จะต้องใช้คำสั่ง ดังนี้

```
float a;
```

```
a = (int)12.423;
```

จะได้ว่า a มีค่าเท่ากับ 12 กระบวนการทำงานจะมีการเปลี่ยนชนิดข้อมูลที่อยู่ใกล้กับตัวดำเนินการให้เป็นชนิดข้อมูลที่ระบุในวงเล็บ แล้วจึงมีการกำหนดค่านั้นใหม่ให้กับ a ทั้งนี้สามารถกำหนดชนิดข้อมูลที่จะเปลี่ยนค่าเป็นชนิดข้อมูลใดๆ ก็ได้ แสดงเพิ่มเติมดังตัวอย่างที่ 4

#### ตัวอย่างที่ 4 การใช้ตัวดำเนินการเปลี่ยนชนิดข้อมูล

```

1 #include <stdio.h>
2 void main( )
3 {
4     float x;
5     x = 2.5 * 2;
6     printf("x value is %0.2f", x);
7     x = (int)2.5 * 2;
8     printf("\nx value is %0.2f", x);
9 }
```

ผลการรันโปรแกรม

```
x value is 5.00
```

```
x value is 4.00
```

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่	9-10
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

### 6. ตัวดำเนินการคอมม่า

ตัวดำเนินการคอมม่าเป็นตัวดำเนินการเพื่อบอกถึงลำดับการทำงาน ช่วยกำหนดการทำงานของนิพจน์หนึ่งๆจะใช้ร่วมกับคำสั่งควบคุมรูปแบบการใช้ตัวดำเนินการคอมม่า คือ นิพจน์ 1, นิพจน์ 2 การทำงานจะทำงานที่นิพจน์ทางซ้ายมือก่อนเสมอแล้วจึงทำงานที่นิพจน์ขวามือ แต่ถ้าใช้คำสั่ง

$$s = (t = 2, t + 3);$$

ลำดับการทำงาน คือ  $t = 2$  หลังจากนั้นจะนำค่าใน  $t$  คือ 2 ไปบวกกับ 3 จะได้ค่า คือ 5 และ นำค่า 5 ไปกำหนดให้กับ  $s$  จะได้ว่า  $t$  มีค่า 2 และ  $s$  มีค่า 5 แต่หากไม่มีเครื่องหมายวงเล็บจะเป็นดังนี้

$$s = t = 2, t + 3;$$

จะได้ว่า  $s$  เท่ากับ 2 และ  $t$  เท่ากับ 2 หลังจากนั้นนำค่า  $t$  ไปบวก 3 ได้ผลลัพธ์ของนิพจน์เป็น  $5 \cdot$  ซึ่งจะใช้นิพจน์ตรรกศาสตร์ในคำสั่งควบคุมแบบต่างๆ

### 7. ตัวดำเนินการความสัมพันธ์

ตัวดำเนินการความสัมพันธ์ ได้แก่ ตัวดำเนินการที่ใช้เปรียบเทียบนิพจน์ 2 นิพจน์ ใช้ร่วมกับคำสั่งควบคุม

ตารางที่ 5.3 ตัวดำเนินการความสัมพันธ์

ตัวดำเนินการ	ความหมาย
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ .

### ตัวอย่างที่ 5 การเปรียบเทียบด้วยตัวดำเนินการความสัมพันธ์

```

1 #include <stdio.h>
2 void main( )
3 {
4     int x, y
5     printf("Enter X : ");
6     scanf("%d", &x);
7     printf("Enter Y : ");
8     scanf("%d", &y);
9     printf("X > Y is %d", x>y);
10 }
```

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>5</b>
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	<b>สอนครั้งที่</b>	9-10
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	8
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	5
	ชื่อหน่วย ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	สอนครั้งที่	9-10
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b> 5
	<b>ชื่อหน่วย</b> ตัวดำเนินการและนิพจน์ทางคณิตศาสตร์	<b>สอนครั้งที่</b> 9-10
		<b>ชั่วโมงรวม</b> 72
		<b>จำนวนชั่วโมง</b> 8
<b>9. การวัดและประเมินผล</b>		
<b>9.1 ก่อนเรียน</b>		
9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย		
<b>9.2 ขณะเรียน</b>		
9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ		
9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน		
<b>9.3 หลังเรียน</b>		
9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%		
<b>10. บันทึกหลังสอน</b>		
<b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b>		
10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก		
10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน		

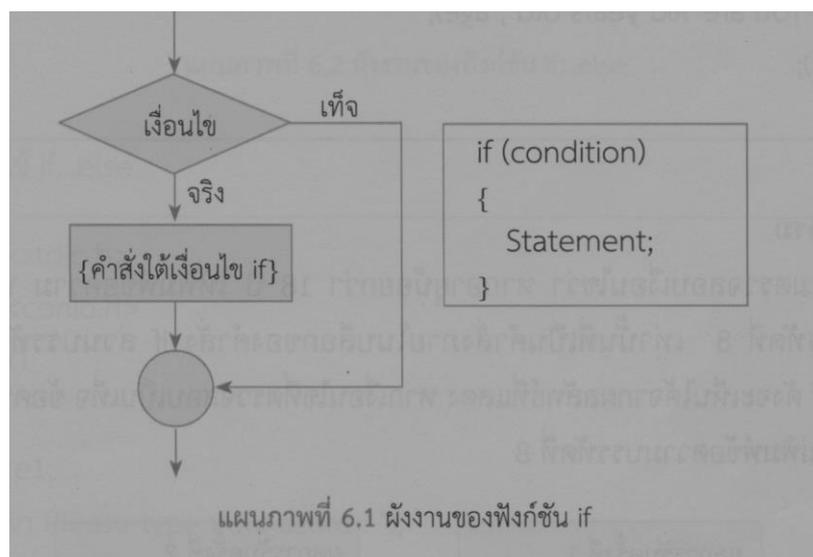
	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่ 11
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4
<p><b>1. สาระสำคัญ</b></p> <p>ในภาษาซีจะใช้ประโยคเงื่อนไข if เพื่อสร้างเงื่อนไข และตรวจสอบเงื่อนไขว่าเป็นจริงหรือเท็จ ทั้งนี้ ประโยค if ดังกล่าว ยังสามารถนำมาใช้งานเพื่อเปรียบเทียบเงื่อนไขอย่างง่าย จนถึงเงื่อนไขที่ซับซ้อนสูงได้</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับการทำฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p>3.1. ด้านความรู้</p> <p>3.1.1 ประโยคเงื่อนไข if อย่างง่ายหรือเรียกว่า if แบบทางเลือกเดียว</p> <p>3.1.2 ประโยคคำสั่งตัดสินใจแบบสองทางเลือกด้วย</p> <p>3.2 ด้านทักษะ</p> <p>-</p> <p>3.3 คุณลักษณะที่พึงประสงค์</p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>		
	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 6

	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	11
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ฟังก์ชัน if

ฟังก์ชัน if (เงื่อนไขทางเลือกเดียว) เป็นการสร้างทางเลือกโดยการตรวจสอบเงื่อนไข ถ้าเงื่อนไขที่กำหนดไว้เป็นจริงให้ประมวลผลตามคำสั่งที่กำหนดไว้ มีรูปแบบการใช้งานดังแผนภาพที่



จากรูปแบบของฟังก์ชัน if มีรายละเอียดดังนี้

if (condition) statement;

condition นิพจน์หรือเงื่อนไขที่กำหนดเพื่อตรวจสอบ

statement ประโยคคำสั่ง



ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	11
	ชั่วโมงรวม	72
	จำนวนชั่วโมง	4

ผลการทำงาน : ส่วน statement จะถูกกระทำถ้าค่านิพจน์ที่อยู่หลังคำว่า if มีค่าเป็นจริง แต่ถ้านิพจน์ดังกล่าวมีค่าเป็นเท็จ โปรแกรมก็จะข้ามส่วนที่เป็น statement นี้ไป

ตัวอย่างที่ 1 โปรแกรมแสดงการทำงานของคำสั่งเงื่อนไข if

```

1 #include <stdio.h>
2 void main()
3 {
4     int age;
5     printf("How old are you = ");
6     scanf("%d", &age);
7     if(age<18)
8     {printf("You are young\n");}
9     printf("You are %d years old", age);
10    getch();
11 }

```

คำอธิบายโปรแกรม

ผลการรันครั้งที่ 1

How old are you = 12  
You are young  
You are 12 years old

ผลการรันครั้งที่ 2

How old are you = 19  
You are 19 years old

ตัวอย่างที่ 1 โปรแกรมแสดงการทำงานของคำสั่งเงื่อนไข if

```

1 #include <stdio.h>
2 void main()
3 {
4     int age;
5     printf("How old are you = ");
6     scanf("%d", &age);
7     if(age<18)
8     {printf("You are young\n");}
9     printf("You are %d years old", age);
10    getch();

```

โปรแกรมตรวจสอบเงื่อนไขว่า หากอายุน้อยกว่า 18 ปี ให้พิมพ์ข้อความ You are young ซึ่งโปรแกรมบรรทัดที่ 8 เท่านั้นที่เป็นคำสั่งภายในบล็อกของคำสั่ง if ส่วนบรรทัดที่ 9 เป็นคำสั่ง นอกบล็อกของ if ดังจะ

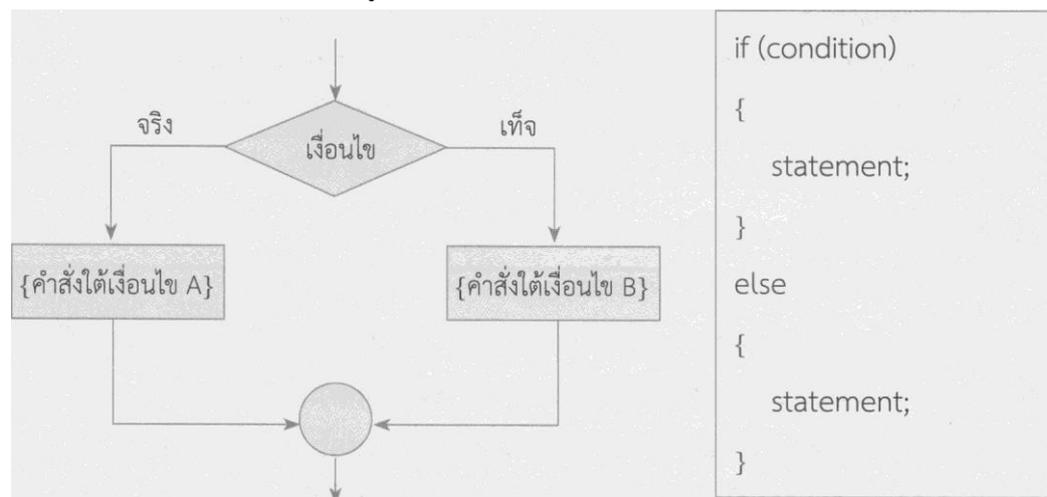


แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

หน่วยที่ 6

ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	11
	ชั่วโมงรวม	72
	จำนวนชั่วโมง	4

ฟังก์ชัน if...else (เงื่อนไขสองทางเลือก) เป็นการสร้างทางเลือกโดยการตรวจสอบเงื่อนไข ถ้า เงื่อนไขที่กำหนดไว้เป็นจริงให้ประมวลผลตามคำสั่งที่ต้องการหลังคำสั่ง if หรือหากเงื่อนไขที่ได้เป็นเท็จก็ให้ประมวลผลหลังคำสั่ง else ซึ่งมีรูปแบบการใช้งานดังนี้



แผนภาพที่ 6.2 ผังงานของฟังก์ชัน if...else

ตัวอย่างที่ 2 การใช้ if...else

คำอธิบายโปรแกรม

เป็นโปรแกรมรับการป้อนค่าคะแนนจากแป้นพิมพ์นำไปเก็บในตัวแปร score1 แล้วนำ score1 มาเปรียบเทียบกับค่า 50 ถ้าค่าที่รับเข้ามามากกว่าหรือเท่ากับ 50 จะได้ผลเป็นจริง โปรแกรมจะทำตามคำสั่งใน if คือ printf("You pass the examination."); แต่ถ้าเปรียบเทียบกับ score1 กับค่า 50 แล้วได้ค่าเป็นเท็จ คือ score1 น้อยกว่า 50 โปรแกรมจะทำตามคำสั่งหลัง else คือ printf("\nYou failed the examination."); แล้วออกจากคำสั่งโดยไม่ทำตามคำสั่งหลัง if

ผลการรันครั้งที่ 1  
Please type your score : 55  
You pass the examination.

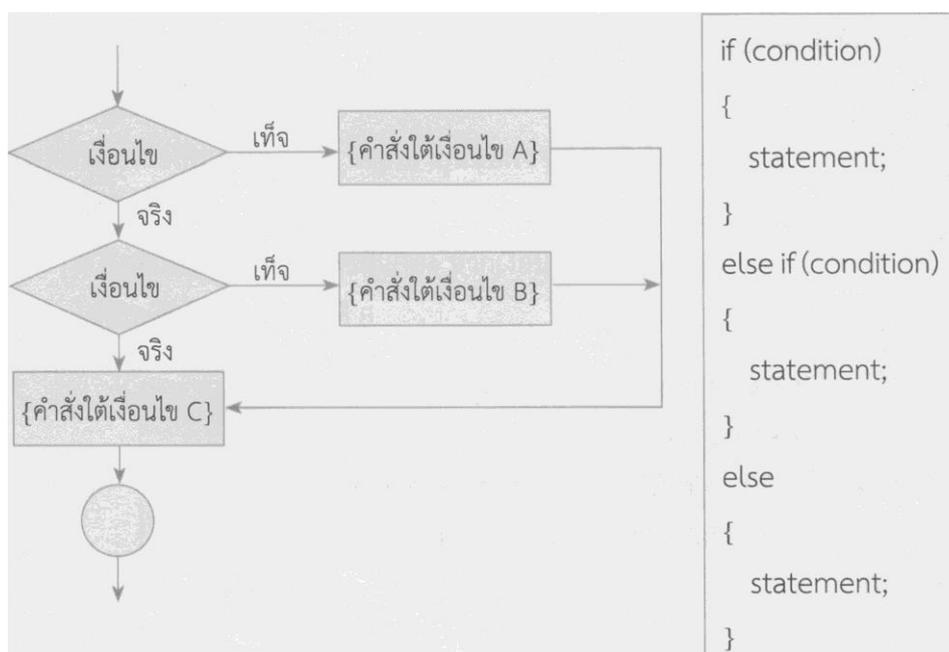
ผลการรันครั้งที่ 2  
Please type your score : 45  
You failed the examination.

### 3. ฟังก์ชัน if...else...if

ฟังก์ชัน if...else...if (เงื่อนไขหลายทางเลือก) เป็นการสร้างทางเลือกโดยการตรวจสอบเงื่อนไข ถ้า มีทางเลือกหลายทาง โดยถ้าเงื่อนไขที่กำหนดไว้เป็นจริงให้ประมวลผลตามคำสั่งที่ต้องการหลังคำสั่ง if ถ้า

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 6</b>
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่ 11
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4

หากไม่ตรงกับเงื่อนไขที่กำหนดไว้ก็จะทำการตรวจสอบเงื่อนไขอื่นๆ ต่อไปหลังคำสั่ง else if เมื่อเงื่อนไข ตรงกับที่กำหนดไว้ก็จะประมวลผลในคำสั่งช่วงนั้น หรือหากเงื่อนไขที่ได้เป็นเท็จก็ให้ประมวลผลหลังคำสั่ง else ตามคำสั่งที่กำหนดไว้ ซึ่งมีรูปแบบการใช้งานดังนี้



แผนภาพที่ 6.3 ผลงานของฟังก์ชัน if...else...if

**ตัวอย่างที่ 3** โปรแกรมการใช้งาน if...else...if

โปรแกรมตัดเกรด กำหนดให้รับค่าคะแนนผ่านทางแป้นพิมพ์และแสดงผลเกรดผ่านทางจอภาพ กำหนดเงื่อนไขการตัดเกรด ดังนี้

คะแนนระหว่าง	80 ถึง	100	ได้เกรด 4
คะแนนระหว่าง	60 ถึง	79	ได้เกรด 3
คะแนนระหว่าง	50 ถึง	59	ได้เกรด 2
คะแนนระหว่าง	40 ถึง	49	ได้เกรด 1
คะแนนระหว่าง	0 ถึง	39	ได้เกรด 0

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่ 11
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4

```

1  #include <stdio.h>
2  #include <conio.h >
3  int ทาลเก0
4  {
5  int score;
6  char grade;
7  printfC'Input Your Score : ');
8  scanf("%d", &score);
9  if (score >= 80) {grade = '4';}
10 else if (score >= 60) {grade = '3';}
11 else if (score >= 50) {grade = '2';}
12 else if (score >= 40) {grade = T;}
13 else {grade = '0';}
14 printfO'Output Your Grade : %c", grade);
15 getchO;
16 return 0;
17 }

```

ผลการรันครั้งที่ 1

Input Your score : 30

Output Your Grade : 0

ผลการรันครั้งที่ 2

Input Your score : 40

Output Your Grade : 1

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	11
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

ผลการรันครั้งที่ 4

Input Your score : 60

Output Your Grade : 3

ผลการรันครั้งที่ 5

Input Your score : 80

Output Your Grade : 4

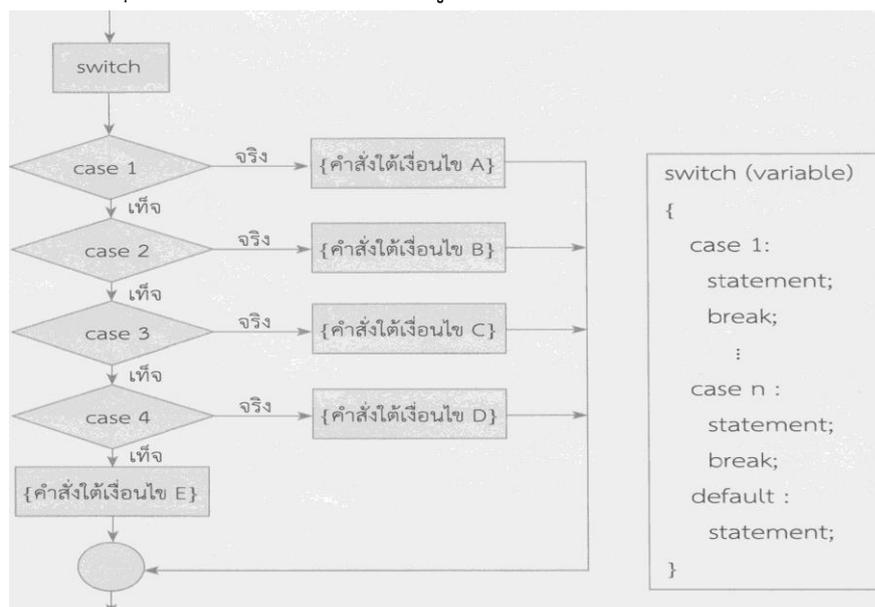
ผลการรันครั้งที่ 3

Input Your score : 50

Output Your Grade : 2

#### 4. ฟังก์ชัน switch...case

ฟังก์ชัน switch...case เป็นการสร้างทางเลือกหลายๆ ทางโดยการตรวจสอบค่าของตัวแปรว่ามีค่า ตรงกับเงื่อนไขหรือ case ใดก็จะประมวลผลตาม case นั้น แต่ถ้าไม่ตรงกับ case ใดๆ เลย ก็จะไปทำงาน ใน default ซึ่งค่าที่ใช้ในการตรวจสอบหรือเปรียบเทียบนั้นจะไม่สามารถใช้ค่าประมาณการหรือมากกว่า น้อยกว่าได้เหมือนกับ if แต่จะต้องใช้ค่าคงที่เป็นตัวอักษรหรือตัวเลขเท่านั้นในการเปรียบเทียบเงื่อนไข ซึ่งการใช้งานฟังก์ชัน switch...case เหมาะสำหรับเงื่อนไขที่มีทางเลือกหลายทาง เนื่องจากถ้าเขียนด้วย if จะทำให้เกิดความยุ่งยากในการเขียนคำสั่งได้ มีรูปแบบการใช้งานดังนี้



	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่ 11
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4
<p>แผนภาพที่ 6.4 ผลงานของฟังก์ชัน switch...case</p> <p>จากรูปแบบของฟังก์ชัน case มีรายละเอียดดังนี้</p> <p>Variable ตัวแปรที่ใช้ในการตรวจสอบค่า</p> <p>Case กรณีหรือทางเลือกที่กำหนดเพื่อให้คอมพิวเตอร์ประมวลผล</p> <p>Default ประโยคคำสั่งเมื่อการตรวจสอบเงื่อนไขไม่ตรงกับ case ใดๆ ที่กำหนดไว้ หรือ เงื่อนไขที่เป็นเท็จ</p> <p>Break หยุดคำสั่งในแต่ละ case หากไม่ใส่ในแต่ละ case โปรแกรมจะทำงานตามคำสั่ง case ต่อไปเรื่อยๆ</p> <p>ตัวอย่างที่ 4 โปรแกรมการใช้งานฟังก์ชัน switch...case เปลี่ยนตัวเลขเป็นเกรด โดยกำหนดให้รับค่า ตัวเลขผ่านทางแป้นพิมพ์และแสดงผลเกรดผ่านทางจอภาพเป็นตัวอักษร</p> <pre> 1 #include &lt;stdio.h&gt; 2 #include &lt;stdlib.h&gt; 3 int main() 4 { 5     int score; 6     printf("Input your key number 1 2 3 4 5: "); 7     scanf("%d", &amp;score); 8     switch(score) { 9         case 5 : printf("Score : %d = A", score); break; 10        case 4 : printf("Score : %d = B", score); break; 11        case 3 : printf("Score : %d = C", score); break; 12        case 2 : printf("Score : %d = D", score); break; 13        case 1 : printf("Score : %d = F", score); break; 14        default: printf("NO key number 1 2 3 4 5"); 15    } 16    return 0; 17 }</pre>		

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่ 11
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4
<p>ผลการรันครั้งที่ 1 Input your key number 1 2 3 4 5 : 5 Score : 5 = A</p> <p>ผลการรันครั้งที่ 2 Input your key number 1 2 3 4 5 : 6 NO key number 1 2 3 4 5</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>6</b>
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	<b>สอนครั้งที่</b>	11
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	4
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	11
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	11
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>7</b>
	ชื่อหน่วย ฟังก์ชันการทำงานแบบทำซ้ำ	<b>สอนครั้งที่</b>	<b>12</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>4</b>
<p><b>1. สาระสำคัญ</b></p> <p>การทำงานแบบวนซ้ำ (Loop)เป็นการนำคำสั่งมาทำงานซ้ำหลายๆ รอบ จะทำงานกี่รอบขึ้นอยู่กับเงื่อนไขที่ผู้เขียนโปรแกรมกำหนดไว้ ซึ่งอาจจะเป็นการกำหนดจำนวนรอบที่แน่นอน เช่น ตั้งใจว่าจะวิ่งรอบสนาม 3 รอบ คือรู้แน่นอนว่าจะทำงานกี่รอบ หรือแบบจำนวนรอบไม่แน่นอน เช่น ตั้งใจว่าจะวิ่งรอบสนามไปเรื่อยๆ เหนื่อยเมื่อไหร่จึงจะหยุดวิ่ง คือไม่แน่ชัดว่าจะทำงานกี่รอบ</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับการฟังก์ชันการทำงานแบบทำซ้ำ</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 เข้าใจหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p>3.1.2 มีทักษะในการวางแผน เขียน ตรวจสอบ และแก้ไขโปรแกรมคอมพิวเตอร์</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	7
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ฟังก์ชัน for

ฟังก์ชันในการวนรอบแบบ for เป็นการวนรอบทำซ้ำจนกว่าค่าของตัวแปรที่ตั้งจะครบตามเงื่อนไขที่ต้องการ โดยการที่จะเพิ่มหรือลดค่าทีละหนึ่งไปเรื่อยๆ ถ้าเงื่อนไขเป็นจริงจะทำตามประโยคคำสั่งภายในเครื่องหมายปีกกา { } เมื่อตรวจสอบเงื่อนไขแล้วพบว่าเงื่อนไขเป็นเท็จก็จะหยุดการทำงานแล้วออกไปทำประโยคคำสั่งหลังเครื่องหมายปีกกาปิด ซึ่งมีรูปแบบการใช้งานดังนี้

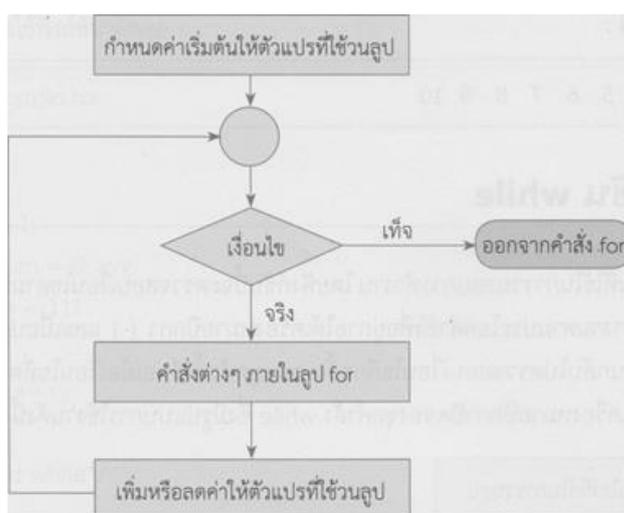
```
for (initialization; condition; increment or decrement)
{
    statement;
    :
}
```

initialization คือ กำหนดค่าเริ่มต้นให้กับตัวแปร

condition คือ นิพจน์ที่ใช้ทดสอบเงื่อนไข ถ้านิพจน์นี้ให้ผลลัพธ์เป็นจริงจะทำตามคำสั่งที่อยู่ภายใน คำสั่ง for

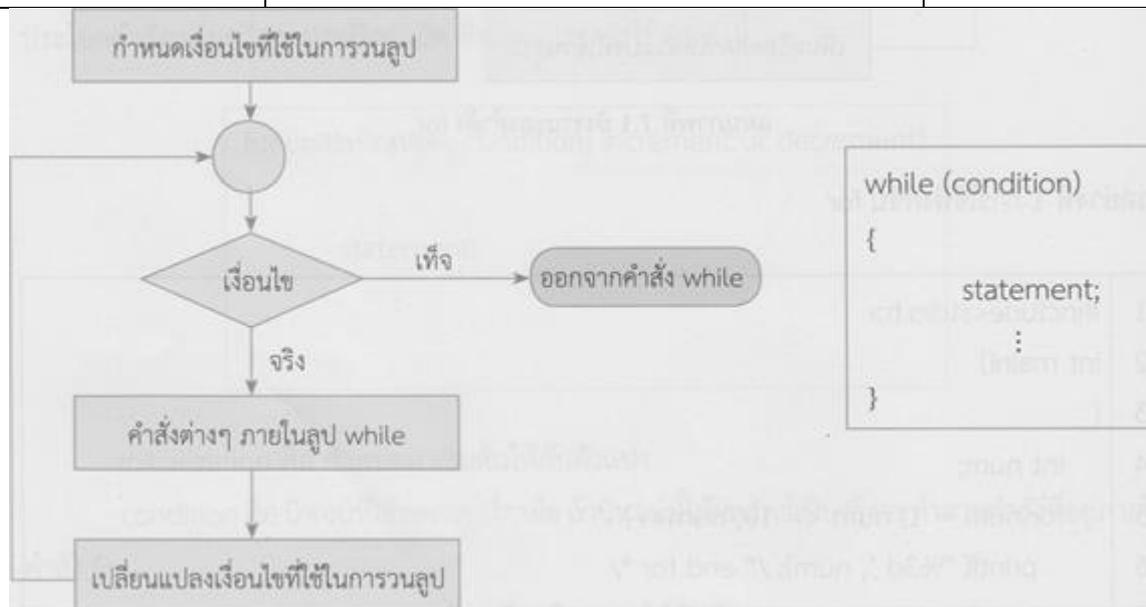
increment or decrement คือ เพิ่มหรือลดค่าให้กับตัวแปร

for (initialization; condition; increment or decrement)



	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	7
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
ตัวอย่างที่ 1 การใช้ฟังก์ชัน for			
1	#include<stdio.h>		
2	int main()		
3	{		
4	int num;		
5	for(num = 1; num <= 10; num++)		
6	printf("%3d ", num); /* end for */		
7	return 0;		
8	}		
<p><b>คำอธิบายโปรแกรม</b></p> <p>จากโปรแกรมตัวอย่างที่ 1 สามารถอธิบายการทำงานของโปรแกรมที่สำคัญๆ ได้ดังนี้ บรรทัดที่ 5 คำสั่ง for(num = 1; num &lt;= 10; num++) จะเริ่มทำงานโดยการกำหนดค่าเริ่มต้นตัวแปร ทนทา เป็น 1 จากนั้นทดสอบเงื่อนไข num &lt;= 10 ว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงจะทำงานตามคำสั่งบรรทัดที่ 6 ไปทำ คำสั่งเพิ่มค่าของตัวแปร ทนทา ในแต่ละรอบของการทำงานจนกว่าค่าของตัวแปรมีความมากกว่า 10 ซึ่ง เป็นเท็จ จะออกจาก loop for</p> <p><b>ผลการรันโปรแกรม</b></p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> 1 2 3 4 5 6 7 8 9 1 0 </div> <p style="text-align: center;"><b>2. ฟังก์ชัน while</b></p> <p>เป็นฟังก์ชันที่ใช้ในการวนรอบการทำงานโดยฟังก์ชันนี้จะตรวจสอบเงื่อนไขตามที่กำหนดก่อน ถ้าเงื่อนไขเป็นจริงจะทำงานตามประโยคคำสั่งที่อยู่ภายใต้เครื่องหมายปีกกา {} และเมื่อปฏิบัติเสร็จสิ้นตามประโยคคำสั่งก็จะวนกลับไปตรวจสอบเงื่อนไขอีกครั้ง จะหยุดทำซ้ำก็ต่อเมื่อเงื่อนไขที่ตรวจสอบเป็นเท็จ แล้วจึงทำคำสั่งหลังเครื่องหมายปีกกาปิดของชุดคำสั่ง while ซึ่งมีรูปแบบการใช้งานดังนี้</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	7
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4



แผนภาพที่ 7.2 ผังงานของคำสั่ง while

conditional คือ นิพจน์ที่ใช้ตรวจสอบเงื่อนไข ถ้านิพจน์นี้ให้ผลลัพธ์เป็นจริงจะทำตามคำสั่งที่อยู่ในคำสั่ง while จนกว่าเงื่อนไขเป็นเท็จจึงออกจากคำสั่ง while ได้

statement คือ ประโยคคำสั่งต่างๆ

ตัวอย่างที่ 2 การใช้ฟังก์ชัน while

```

1 #include<stdio.h>
2 int main()
3 {
4     int n = 1;
5     float sum = 0, avg;
6     while(n < 11)
7     {
8         sum += n;
9         n++;
10    } /* end while*/
11    n--;
12    avg = sum/n;
13    printf("N = %d, Sum = %.2f\n", n, sum);
14    printf("Average = %.2f, avg);

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	7
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

### คำอธิบายโปรแกรม

บรรทัดที่ 6 คำสั่ง `while(n < 11)` ควบคุมการทำงานของโปรแกรมให้ทำงานซ้ำกันโดยการทำงานของคำสั่ง `while` จะตรวจสอบเงื่อนไข `n < 11` ว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงจะทำงานตามคำสั่งที่อยู่ใน `loop while` คือ บรรทัดที่ 8 และ 9 จะคำนวณค่าสะสมของตัวแปร `sum` และเพิ่มค่า `n` ทีละ 1 ตามลำดับ จากนั้นจะกลับมาตรวจสอบเงื่อนไขใหม่ ทำอย่างนี้ซ้ำๆ จนกว่าจะตรวจสอบเงื่อนไขเป็นเท็จ ซึ่ง `n` นั้นมากกว่าหรือเท่ากับ จึงออกจาก `loop while` แล้วไปทำคำสั่งบรรทัดที่ 11 ถึงคำสั่งบรรทัดที่ 16 เป็นคำสั่งที่โปรแกรมจะทำงานหลังจากตรวจสอบเงื่อนไข `while` คือ บรรทัดที่ 11 เป็นการลดค่าตัวแปร `n` ลง 1 แล้วคำนวณค่าเฉลี่ยเก็บไว้ที่ตัวแปร `avg` และพิมพ์ค่าตัวแปร `n`, `sum` และ `avg` แสดงที่จอภาพ พร้อมกับพิมพ์ข้อความให้กดคีย์ใดๆ เพื่อกลับสู่โปรแกรม

### ผลการรันโปรแกรม :

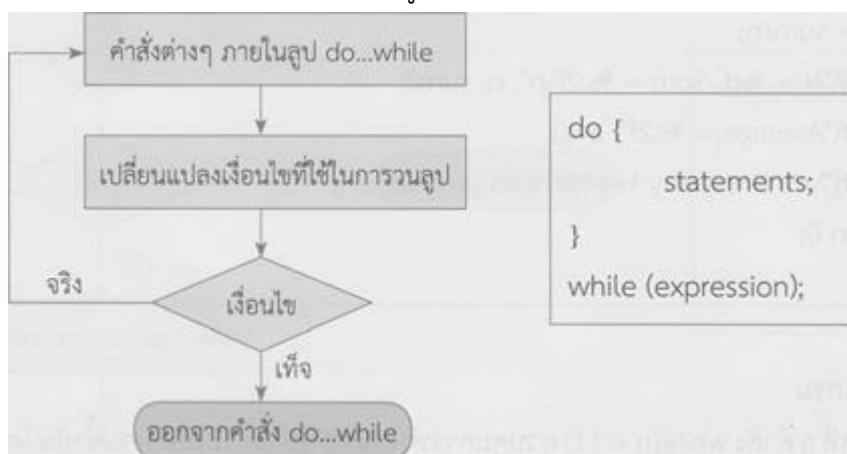
`N = 10, Sum 55.00`

`Average = 5.50`

`Press any key back to program...`

### 3. ฟังก์ชัน `do...while`

ฟังก์ชัน `do...while` เป็นฟังก์ชันที่ใช้ในการวนรอบการทำงาน โดยฟังก์ชันนี้จะทำงานตามประโยค คำสั่งที่ระบุภายในเครื่องหมายปีกกา `{}` ก่อน แล้วจึงตรวจสอบเงื่อนไขตามที่กำหนด ถ้าเงื่อนไขเป็นจริงจะวนไปทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ ซึ่งมีรูปแบบการใช้งานดังนี้



	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b> 7	
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p style="text-align: center;">แผนภาพที่ 7.3 ผังงานของคำสั่ง do...while conditional คือ นิพจน์ที่ใช้ทดสอบเงื่อนไข ถ้านิพจน์นี้ให้ผลลัพธ์เป็นจริงจะทำตามคำสั่งที่อยู่ ภายในคำสั่ง while จนกว่าเงื่อนไขเป็นเท็จจึงออกจากคำสั่ง while ได้ statement คือ ประโยคคำสั่งต่างๆ</p> <p style="text-align: center;"><b>do {statement;} while (conditional);</b></p> <p>ข้อควรระวังในการใช้คำสั่ง do...while คือ ควรระวังเงื่อนไขของคำสั่ง do...while ต้องพิจารณา ให้ดี ว่ามีทั้งกรณีที่เป็นจริงและเท็จอยู่หรือไม่ ถ้ามีอยู่ทั้ง 2 กรณี แสดงว่าใช้คำสั่งนี้ได้ถูกต้องตามไวยากรณ์ ของคำสั่งนี้ ถ้ามีเฉพาะกรณีที่เงื่อนไขเป็นจริงเท่านั้นแสดงว่าเกิดลักษณะการทำงานวนลูป (Looping) ไม่มีทางออกจากคำสั่งนี้ในทำนองกลับกันถ้าเงื่อนไขเป็นเท็จอย่างเดียว จะทำคำสั่ง do...while เพียงครั้งเดียว</p> <p><b>ตัวอย่างที่ 3</b> การใช้คำสั่ง do...while เพื่อวนลูปแสดง Main Menu ให้ผู้ใช้เลือกเมนูไปเรื่อยๆ จนเลือก เมนูที่ 5 จะออกจากโปรแกรม</p> <pre> 1 #include &lt;stdio.h&gt; 2 int main() 3 { 4     char choice; 5     do { 6         printf("\n\n***** MAIN MENU *****\n"); 7         printf("*****\n\n"); 8         printf("1. Create New File\n"); 9         printf("2. Use an Old File\n"); 10        printf("3. Edit data record in File\n"); 11        printf("4. Append data record in File\n"); 12        printf("5. Exit Program\n"); 13        printf("Enter your choice (1, 2, 3, 4, 5) : "); 14        choice = getche(); 15    } while(choice != '5'); 16    return 0; 17 }</pre>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>7</b>
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	<b>สอนครั้งที่</b>	<b>12</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>4</b>

### คำอธิบายโปรแกรม

บรรทัดที่ 5 คำสั่ง do จะควบคุมการทำงานของโปรแกรม ให้ทำตามคำสั่งที่อยู่ภายใน loop do คือ คำสั่งภายในเครื่องหมาย { } บรรทัดที่ 6 ถึง 14 ซึ่งจะทำอย่างน้อย 1 รอบ หลังจากนั้นค่อยไป ตรวจสอบเงื่อนไข while ในบรรทัดที่ 15 ถ้าเป็นจริงจะกลับมาทำคำสั่งใน loop อีกครั้ง แต่ถ้าเป็นเท็จก็ จะออกจากโปรแกรม และสิ้นสุดการทำงาน

### ผลการรันโปรแกรม :

```

***** MAIN MENU *****
*****
1. Create New File
2. Use an Old File
3. Edit data record in file
4. Append data record in File
5. Exit Program
Enter your choice (1, 2, 3, 4, 5) :

```

### 4. ฟังก์ชัน break

ฟังก์ชัน break เป็นฟังก์ชันที่ใช้ในการกำหนดให้ออกจากการทำงานของฟังก์ชันวนรอบการทำงาน ซึ่งรูปแบบของการใช้งาน มีดังนี้

#### 4.1 รูปแบบฟังก์ชัน break ในลูป for มีดังนี้

```

for (initialization; condition; increment or decrement)
{
    statement;
    if (condition) break;
}

```

#### 4.2 รูปแบบฟังก์ชัน break ในลูป while มีดังนี้

```

while (condition)
{
    statement;
    if (condition) break;
}

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	7
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

แผนภาพที่ 6.4 ผลงานของฟังก์ชัน switch...case

#### 4.3 รูปแบบฟังก์ชัน break ในลูป do...while มีดังนี้

ตัวอย่างที่

```
do
{
    statement;
    if (condition) break;
}
while (condition);
```

4 การใช้ฟังก์ชัน break ในลูป for

```
1 #include <stdio.h>
2 int main()
3 {
4     int j;
5     for (j = 1; j <=20; j++) {
6         printf("%d\t", j);
7         if (j ==10) break;
8     }
9     printf("\n\nPress any key back to program...");
10    Getch();
11    return 0;
12 }
```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	7
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>คำอธิบายโปรแกรม</b></p> <p>บรรทัดที่ 5 คำสั่ง for (j = 1; j &lt;= 20; j++) จะให้โปรแกรมวนพิมพ์ค่าของตัวแปร j ตั้งแต่ 1 ถึง 20 โดยแต่ละค่าให้ห่างกัน 1 Tab บรรทัดที่ 7 คำสั่ง if (j == 10) break; จะตรวจสอบค่าตัวแปร j ว่า เท่ากับ 10 หรือไม่ ถ้าเท่ากับ 10 ให้หยุดการทำงานและออกจาก loop for ดังนั้นโปรแกรมนี้จะได้พิมพ์ ตัวเลข 1 ถึง 20 เพราะพบคำสั่ง break แต่จะพิมพ์ตัวเลข 1 ถึง 10 เท่านั้น</p> <p><b>ผลการรันโปรแกรม :</b></p> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 10px auto;"> <pre>1      2      3      4      5      6      7      8      9      10 Press any key back to program...</pre> </div> <p><b>5.ฟังก์ชัน continue</b></p> <p>ฟังก์ชัน continue เป็นฟังก์ชันที่ใช้งานตรงข้ามกับฟังก์ชัน break โดยจะบังคับให้โปรแกรมกระโดดข้ามไปยังรอบต่อไปในการทำงานตามเงื่อนไขที่กำหนดภายในลูป มีรูปแบบการทำงานตามลูปดังนี้</p> <p><b>ตัวอย่างที่ 5</b> การใช้ฟังก์ชัน continue ในลูป for</p> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 10px auto;"> <pre>1 #include &lt;stdio.h&gt; 2 int main() 3 { 4     int j; 5     for (j = 1; j &lt;= 20; j++){ 6         printf("%o/od\t",j); 7         if (j==5) {j = j+10; continue; } 8     } 9     printf("\n\nPress any key back to program..."); 10    return 0; 11 }</pre> </div>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>7</b>
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	<b>สอนครั้งที่</b>	<b>12</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>4</b>
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	6
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	สอนครั้งที่	12
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>6</b>
	ชื่อหน่วย ฟังก์ชันตรวจสอบเงื่อนไขแบบทางเลือก	<b>สอนครั้งที่</b>	<b>12</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>4</b>
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

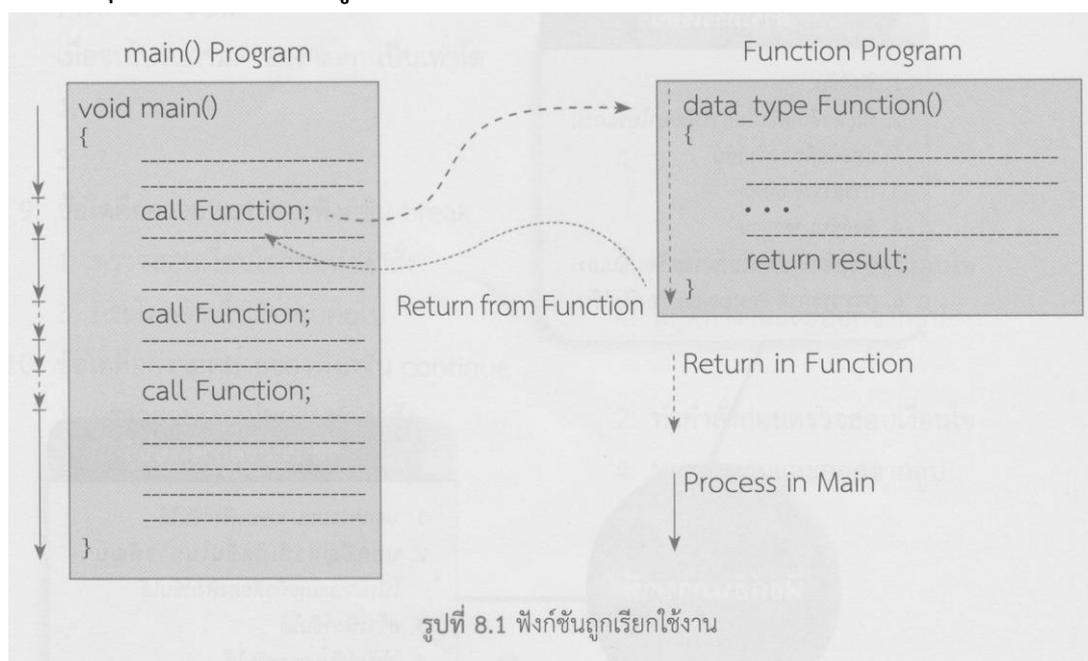
	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>8</b>
	ชื่อหน่วย ฟังก์ชันภาษาซี	<b>สอนครั้งที่</b>	13-14
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	8
<p><b>1. สาระสำคัญ</b></p> <p>ฟังก์ชัน คือส่วนของโปรแกรมที่ถูกเขียนขึ้นเพื่อทำหน้าที่เฉพาะของมัน ฟังก์ชันในภาษา C นั้นจะมีฟังก์ชันที่มากับภาษา และฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง ในบทนี้เราจะพูดถึงฟังก์ชันที่จะสร้างขึ้นเองเป็นส่วนมาก</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับฟังก์ชันภาษาซี</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรมภาษาอื่น ๆ</p> <p>3.1.2 เขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่	13-14
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8

#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ฟังก์ชัน

ฟังก์ชัน คือ ชุดของการทำงาน ที่ถูกเขียนขึ้นให้โปรแกรมเมอร์สามารถเรียกใช้งานได้ง่าย



#### 2. ปัญหาที่เกิดขึ้นในการพัฒนาโปรแกรมแล ข้อดีของฟังก์ชัน

##### 2.1 ปัญหาที่เกิดขึ้นในการพัฒนาโปรแกรมที่ซับซ้อน มีดังนี้

2.1.1 โปรแกรมเมอร์ไม่สามารถทราบการทำงานของระบบโดยละเอียดได้ เช่น ไม่ทราบ กระบวนการส่งข้อมูลผ่านเครือข่าย แต่ต้องเขียนโปรแกรมเพื่อเชื่อมต่อระบบเครือข่าย เป็นต้น

2.1.2 โปรแกรมเมอร์ไม่สามารถทราบขั้นตอนการทำงานของคอมพิวเตอร์ทั้งหมดได้ เช่น ทำอย่างไร ตัวอักษรจึงปรากฏในหน้าจอภาพได้ เป็นต้น

2.1.3 โปรแกรมบางโปรแกรมมีการทำงานที่ซับซ้อน และการทำงานซับซ้อนนั้นถูกเรียกใช้ งานบ่อยครั้ง เช่น การหาผลลัพธ์ทางวิทยาศาสตร์ การวิเคราะห์ข้อมูลขนาดใหญ่ เป็นต้น

##### 2.2 ข้อดีของฟังก์ชัน มีดังนี้

2.2.1 ทำให้โปรแกรมเมอร์สามารถพัฒนาโปรแกรมได้โดยง่าย โดยโปรแกรมเมอร์ไม่จำเป็นต้องทราบว่าการทำงานของฟังก์ชันทำงานอย่างไร ทราบเพียงผลลัพธ์ของการทำงานเท่านั้น

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 8</b>
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p>2.2.2 โปรแกรมเมอร์สามารถเขียนโปรแกรมให้มีการทำงานที่ซับซ้อนได้ โดยไม่จำเป็นต้อง เขียนโปรแกรม ส่วนที่ซับซ้อนนั้นหลายๆ ครั้ง</p> <p>2.2.3 โปรแกรมเมอร์สามารถออกแบบฟังก์ชันได้ตามความต้องการ</p> <p><b>3. การสร้างฟังก์ชัน</b></p> <p>การสร้างฟังก์ชันทั้งฟังก์ชันมาตรฐานและฟังก์ชันที่ผู้ใช้สร้างขึ้นรูปแบบของฟังก์ชันโปรแกรมภาษาซี จะต้องมียูปร่างแบบนี้</p> <p><b>3.1 ประเภทของข้อมูล</b> หมายถึง ประเภทของข้อมูลที่ส่งกลับ ส่วนนี้จะมีหรือไม่มีก็ได้ขึ้นอยู่กับ การทำงานของฟังก์ชัน ถ้าหากฟังก์ชันไม่มีการส่งค่ากลับ ส่วนนี้จะประกาศเป็น void</p> <p><b>3.2 ชื่อฟังก์ชัน</b> เป็นชื่อของฟังก์ชันที่สร้างขึ้น สามารถเรียกใช้งานผ่านชื่อนี้ได้ โดยการตั้งชื่อจะ ต้องเป็นไปตามกฎการตั้งชื่อในภาษาซี</p> <p><b>3.3 Parameter</b> เป็นตัวแปรที่ใช้รับค่าผ่านเข้ามาในฟังก์ชัน ส่วนนี้จะมีหรือไม่มีก็ได้ขึ้นอยู่กับ การทำงานของฟังก์ชัน</p> <p><b>3.4 return</b> ใช้ในการส่งค่ากลับ ส่วนนี้จะมีหรือไม่มีก็ได้ สำหรับในกรณีที่ฟังก์ชันมีการส่งค่ากลับ จะต้องเขียนค่าที่ส่งกลับตามหลัง return</p> <p><b>•โครงสร้างของฟังก์ชัน (Structure of Functions)</b></p> <pre style="background-color: #f0f0f0; padding: 10px;"> type function_name(type 1 arg 1, type 2 arg 2, ..., type n arg n) {     local variable declaration;     statement 1;     statement 2;     :     statement n;     return(value); } </pre> <p style="text-align: right;">เมื่อ type</p> <p>คือ ชนิดของฟังก์ชันหรือชนิดของข้อมูล เช่น int, float, char, double, void ที่ส่งไป ให้ตำแหน่งที่เรียกใช้ ถ้า ไม่มีการส่งค่ากลับไปให้ตำแหน่งที่เรียกใช้จะใช้ void ถ้าไม่มีการกำหนดจะได้ข้อมูล ที่ส่งกลับมาเป็น int เสมอ</p> <p>function_name คือ ชื่อของฟังก์ชัน ควรตั้งชื่อให้สื่อความหมายถึงการทำงานของฟังก์ชัน และ ต้องเป็นไปตามกฎเกณฑ์การตั้งชื่อตัวแปรในภาษาซี</p> <p>type 1 arg 1, type 2 arg 2,..., type n arg n เป็นชนิดและชื่อของ Argument หรือ Parameter ค่าที่ 1 ถึง N ที่จะรับมาจากผู้เรียกใช้ฟังก์ชัน ถ้าเป็นฟังก์ชันที่ไม่มีการรับและส่งค่า Parameter จะใช้คำว่า void { คือ จุดเริ่มต้นของฟังก์ชัน</p>		

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 8</b>
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8
<p>location variable declaration คือ ส่วนที่มีการประกาศชนิดและชื่อของตัวแปรที่จะใช้เฉพาะ ในฟังก์ชัน โดยส่วนที่อยู่นอกฟังก์ชันจะไม่สามารถเรียกใช้ตัวแปรเหล่านี้ได้</p> <p>statement 1; statement 2;... statement n; เป็นคำสั่งที่ใช้งานในฟังก์ชัน</p> <p>return(value); คือ คำสั่งที่ใช้ส่งข้อมูลหรือค่าที่เป็นผลลัพธ์จากการประมวลผลในฟังก์ชันไปให้ ผู้เรียกใช้ ฟังก์ชันโดยชนิดของข้อมูลที่ส่งต้องเป็นชนิดเดียวกับชนิดของฟังก์ชันหรือชนิดของข้อมูลของฟังก์ชัน ถ้าเป็น ฟังก์ชันที่ไม่มีการส่งค่า (void) จะไม่ต้องใช้คำสั่ง return</p> <p>} คือ ส่วนสิ้นสุดของฟังก์ชัน</p> <p>ฟังก์ชันในภาษาซีมี 2 ชนิด คือ ฟังก์ชันมาตรฐาน (Standard Function) กับฟังก์ชันที่ผู้เขียน โปรแกรม เขียนขึ้นเอง (User Defined Function)</p> <p><b>4. ฟังก์ชันมาตรฐาน</b></p> <p>เป็นฟังก์ชันที่ผู้ผลิตคอมพิวเตอร์เขียนขึ้นเพื่อนำไปใช้ในการเขียนโปรแกรมได้สะดวกและง่ายขึ้นบางครั้ง อาจเรียกว่า Library Functions ปกติฟังก์ชันเหล่านี้จะจัดเก็บไว้ใน Header File ดังนั้นผู้ใช้ จะต้องรู้ว่า ฟังก์ชันนั้นอยู่ใน Header File ไต จึงจะนำไปเรียกใช้ในส่วนต้นของโปรแกรมด้วย include &lt;header file.h&gt; ได้ เช่น #include &lt;stdio.h&gt; เป็นต้น นอกจากนี้ยังมีฟังก์ชันอีกมากมาย เช่น ฟังก์ชัน ทางคณิตศาสตร์ ฟังก์ชันเกี่ยวกับตัวอักษร เป็นต้น</p> <p><b>ตัวอย่างที่ 1</b> การเรียกฟังก์ชันทางคณิตศาสตร์ที่อยู่ในคลังโปรแกรม (Library File) math.h มาใช้</p> <pre> 1 #include &lt;stdio.h&gt; 2 #include &lt;math.h&gt; 3 int main() 4 { 5     float a = 50; 6     printf("%0.2f\n", sqrt(a)); 7     printf("%0.2f\n", sqrt(36)); 8 }</pre>		

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

7.07

6.00

ผลการรันโปรแกรม :

### 5. ฟังก์ชันที่ผู้เขียนโปรแกรมเขียนขึ้นเอง

ฟังก์ชันที่เขียนขึ้นบางครั้งเรียกว่าโปรแกรมย่อย คือ ส่วนของโปรแกรมที่เขียนขึ้นเพื่อให้สามารถทำงานได้อย่างใดอย่างหนึ่งตามความต้องการ สำหรับผู้เขียนโปรแกรมภาษาซี นิยมเรียกโปรแกรมย่อยว่า “ฟังก์ชัน” ส่วนผู้ที่เขียนโปรแกรมภาษาปาสคาล โคบอล ฟอ์แทรน เบสิก นิยมเรียกว่า “โปรแกรมย่อย” สามารถจำแนกฟังก์ชันที่เขียนขึ้นตามลักษณะการส่งค่าไปและรับค่ากลับได้ 4 แบบ คือ

- 1) ฟังก์ชันที่ไม่มีการรับส่งค่า
- 2) ฟังก์ชันที่มีการรับค่ากลับแต่ไม่มีการส่งค่าไป
- 3) ฟังก์ชันที่ไม่มีการรับค่ากลับแต่มีการส่งค่าไป
- 4) ฟังก์ชันที่มีทั้งการรับค่ากลับและส่งค่าไป

ซึ่งฟังก์ชันแต่ละแบบก็เหมาะกับงานแต่ละอย่าง ดังนั้นผู้เขียนฟังก์ชันจึงจำเป็นที่จะต้องศึกษา ทำความเข้าใจฟังก์ชันแต่ละแบบ เพื่อจะได้มาประยุกต์ใช้กับงานได้อย่างเหมาะสม

**5.1 ฟังก์ชันที่ไม่มีการรับส่งค่า** เป็นฟังก์ชันที่ไม่มีการรับค่าเข้ามาในฟังก์ชัน และไม่มีการส่งค่า กลับออกไปจากฟังก์ชัน

โครงสร้างของฟังก์ชันที่ไม่มีการรับส่งค่า

```

#include <stdio.h>
#include <conio.h>
void function_name(void);
int main()
{
    ...
    function_name();
    ...
    return 0;
}

void function_name()
{
    statement 1;
    statement 2;
    :
    statement n;
}

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

ตัวอย่างที่ 2 ฟังก์ชันที่ไม่มีการรับส่งค่า

```

1 #include <stdio.h>
2 void PrintHello(void);
3 int main()
4 {
5     PrintHello();
6     printf("Hello, in function main.\n");
7     PrintHello();
8     return 0;
9 }
10 void PrintHello(void)
11 {
12     printf("Hello, in function PrintHello ..\n\n");
13 }

```

ผลการรันโปรแกรม :

```

Hello, in function PrintHello ..
Hello, in function main.
Hello, in function PrintHello ...

```

5.2 ฟังก์ชันที่มีการรับค่ากลับแต่ไม่มีการส่งค่าไป การเรียกใช้ฟังก์ชันประเภทนี้ต้องรับค่า Argument หรือ Parameter ใดๆ ไปให้ฟังก์ชัน แต่ฟังก์ชันจะไม่มีค่าส่งค่าใดกลับมาให้ ส่วนมากจะใช้ แสดงรายละเอียดหรือ ข้อความที่ต้องแสดงบ่อยๆ

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 8</b>
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

### คำอธิบายโปรแกรม

บรรทัดที่ 5 คำสั่ง do จะควบคุมการทำงานของโปรแกรม ให้ทำตามคำสั่งที่อยู่ภายใน loop do คือ คำสั่งภายในเครื่องหมาย { } บรรทัดที่ 6 ถึง 14 ซึ่งจะทำอย่างน้อย 1 รอบ หลังจากนั้นค่อยไป ตรวจสอบเงื่อนไข while ในบรรทัดที่ 15 ถ้าเป็นจริงจะกลับมาทำคำสั่งใน loop อีกครั้ง แต่ถ้าเป็นเท็จก็ จะออกจากโปรแกรม และสิ้นสุดการทำงาน

### ผลการรันโปรแกรม :

```

***** MAIN MENU *****
*****
1. Create New File
2. Use an Old File
3. Edit data record in file
4. Append data record in File
5. Exit Program
Enter your choice (1, 2, 3, 4, 5) :

```

### 4. ฟังก์ชัน break

ฟังก์ชัน break เป็นฟังก์ชันที่ใช้ในการกำหนดให้ออกจากการทำงานของฟังก์ชันวนรอบการทำงาน ซึ่งรูปแบบของการใช้งาน มีดังนี้

#### 4.1 รูปแบบฟังก์ชัน break ในลูป for มีดังนี้

```

for (initialization; condition; increment or decrement)
{
    statement;
    if (condition) break;
}

```

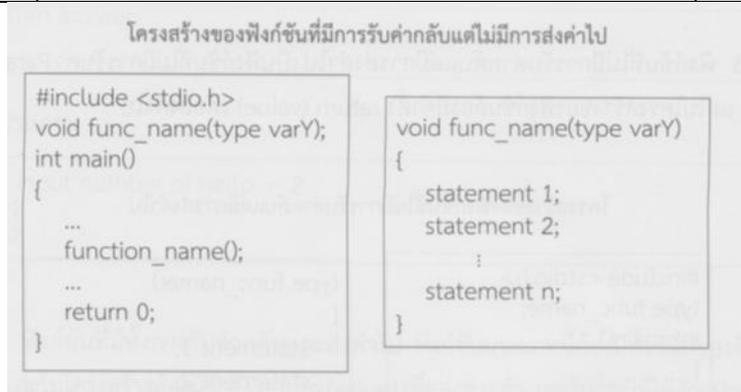
#### 4.2 รูปแบบฟังก์ชัน break ในลูป while มีดังนี้

```

while (condition)
{
    statement;
    if (condition) break;
}

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8



ตัวอย่างที่ 3 ฟังก์ชันที่มีการรับค่ากลับแต่ไม่มีการส่งค่าไป

<pre> 1 #include &lt;stdio.h&gt; 2 void PrintHello(int); 3 int main() 4 { 5     int n; 6     printf("Input number of hello : "); 7     scanf("%d", &amp;n); 8     PrintHello(n); 9     return 0; 10 } 11 void PrintHello(int i) 12 { 13     int count; 14     for (count = 1; count &lt;= i; count++) 15         printf("%d HELLO", count); 16 } </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**ผลการรันโปรแกรม :**

```

Input number of hello: 2
1 HELLO
2 HELLO

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

5.3 ฟังก์ชันที่ไม่มีการรับค่ากลับแต่มีการส่งค่าไป เป็นฟังก์ชันที่ไม่มีการรับค่า Parameter หรือ Argument แต่ในโครงสร้างของฟังก์ชันต้องมีคำสั่ง return (value) เพื่อส่งค่าไป

โครงสร้างของฟังก์ชันที่ไม่มีการรับค่ากลับแต่มีการส่งค่าไป

```

#include <stdio.h>
type func_name;
int main()
{
    ...
    var = func_name();
    ...
    return 0;
}

type func_name()
{
    statement 1;
    statement 2;
    :
    statement n;
    return (varZ);
}

```

ตัวอย่างที่ 4 ฟังก์ชันที่ไม่มีการรับค่ากลับแต่มีการส่งค่าไป

```

1 #include <stdio.h>
2 int RoundInput(void);
3 int main(){
4     int i, round;
5     round = RoundInput();
6     for(i = 1; i <= round; i++) printf("hello # %d\n", i);
7     return 0;
8 }
9 int RoundInput(void){
10    int answer;
11    printf("Please input number of hello : ");
12    scanf("%d", &answer);
13    return answer;
14}

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่ 8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่ 13-14
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 8

ผลการรันโปรแกรม :

Please input number of hello : 2

hello #1

hello #2

5.4 ฟังก์ชันที่มีทั้งการรับค่ากลับและส่งค่าไป ฟังก์ชันประเภทนี้ต้องมีทั้งชนิดของฟังก์ชันและ Argument และในโครงสร้างต้องมีคำสั่ง return (value) เพื่อส่งค่ากลับ ยกเว้นกรณีที่ใช้ตัวแปรพอยน์เตอร์ มาเป็นตัวรับส่งค่า ก็อาจไม่ต้องมีคำสั่ง return

โครงสร้างของฟังก์ชันที่มีทั้งการรับค่ากลับและส่งค่าไป

```

#include <stdio.h>
#include <coino.h>
type func_name(type);
int main()
{
    ...
    var = func_name(varX);
    ...
    return 0;
}

type func_name(type varY)
{
    statement 1;
    statement 2;
    :
    statement n;
    return (varZ);
}

```

ตัวอย่างที่ 5 ฟังก์ชันที่มีทั้งการรับค่ากลับและส่งค่าไป

```

1 #include <stdio.h>
2 float CircleArea(int);
3 int main(){
4     int radius;
5     printf("Input Radius : "); scanf("%d", &radius);
6     printf("Circle Area = %f\n", CircleArea(radius));
7     return 0;
8 }
9 float CircleArea(int rad){
10    float answer = 0;
11    answer = 22.0/7.0*rad*rad;
12    return answer;
13 }

```

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>8</b>
	ชื่อหน่วย ฟังก์ชันภาษาซี	<b>สอนครั้งที่</b>	13-14
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	8
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูชักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่	13-14
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	8
	ชื่อหน่วย ฟังก์ชันภาษาซี	สอนครั้งที่	13-14
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	8
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 9</b>
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่ 15
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4
<p><b>1. สาระสำคัญ</b></p> <p>โครงสร้างข้อมูลแบบอาร์เรย์ เป็นโครงสร้างข้อมูลประเภทเชิงเส้นมีรูปแบบการจัดเก็บ ข้อมูลแบบเรียงลำดับต่อเนื่องกัน โครงสร้างข้อมูลแบบอาร์เรย์ภายในตัวแปรเดียวกันสามารถเก็บ ค่าได้มากกว่า 1 ค่า แต่ละค่ามีขนาดเท่ากันและเป็นข้อมูลชนิดเดียวกัน การเข้าถึงข้อมูลในอาร์เรย์ ท าได้โดยการใช้เลขดัชนี (Index) หรือซบสคริปต์(Subscript) เป็ นตัวอ้างอิงตำแหน่งสมาชิกใน อาร์เรย์ ทำให้การเขียนโปรแกรม เพื่อเข้าถึงข้อมูลทำได้อย่างรวดเร็วและการอ้างอิงข้อมูลทำได้ง่าย</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. อธิบายความหมายของอาร์เรย์ได้</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 อธิบายความหมายของอาร์เรย์ได้</p> <p>3.1.2 บอกคุณลักษณะของอาร์เรย์ได้</p> <p>3.1.3 อ้างอิงต าแหน่งสมาชิกของอาร์เรย์แต่ละมิติได้</p> <p>3.1.4 อธิบายขอบเขตของอาร์เรย์ได้</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>		

		<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	
		ชื่อหน่วย อาร์เรย์	<b>หน่วยที่ 9</b>
			สอนครั้งที่ 15
			ชั่วโมงรวม 72
		จำนวนชั่วโมง 4	

## 1. ประเภทของตัวแปรชุด

อาจแบ่งตามลักษณะของจำนวนตัวเลขของดัชนี คือ

1.1 ตัวแปรชุด 1 มิติ (One-dimensional Arrays หรือ Single-dimensional Array) เป็น ตัวแปรชุดที่มีตัวเลขแสดงขนาดเป็นเลขตัวเดียว เช่น word[20], num[25], x[15] เป็นต้น

1.2 ตัวแปรชุดหลายมิติ (Multi-dimensional Arrays) เป็นตัวแปรชุดที่ชื่อมีตัวเลขแสดงขนาดเป็นตัวเลขหลายตัว ที่นิยมใช้กันมี 2 มิติ กับ 3 มิติ

1.2.1 ตัวแปรชุด 2 มิติ มีเลขแสดงขนาด 2 ตัว เช่น a[3][5], name[5][6] เป็นต้น

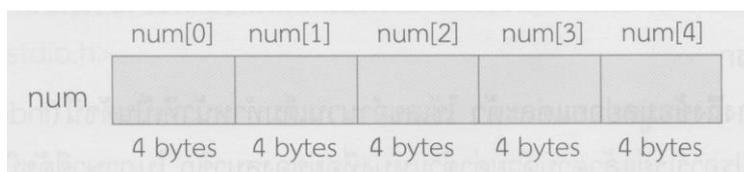
1.2.2 ตัวแปรชุด 3 มิติ มีเลขแสดงขนาด 3 ตัว เช่น a[3][5][6], name[5][6][8] เป็นต้น

## 2. อาร์เรย์ 1 มิติ

2.1 รูปแบบการประกาศตัวแปร มีดังนี้

ชนิดข้อมูล ชื่อตัวแปร [ขนาดของอาร์เรย์]

เช่น float num [5];



```
int    a [25];
char   name[50];
long   value[100];
```

โดยที่ขนาดของอาร์เรย์ สามารถกำหนดเป็นตัวเลขจำนวนเต็มหรือค่าคงที่ก็ได้ แต่เป็นตัวแปรไม่ได้

2.2 การกำหนดค่าเริ่มต้นให้กับอาร์เรย์ 1 มิติ มีดังนี้

2.2.1 สามารถกำหนดค่าเริ่มต้นให้กับอาร์เรย์ได้ตั้งแต่ตอนประกาศตัวแปร

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
		สอนครั้งที่	15
	ชื่อหน่วย อาร์เรย์	ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

2.2.2 ค่าที่กำหนดต้องอยู่ในเครื่องหมาย { } และถ้ามีมากกว่า 1 ค่า ต้องแยกจากกันด้วย เครื่องหมาย , (Comma)

เช่น `int a[5] = {10, 20, 30, 40, 50};`

	a[0]	a[1]	a[2]	a[3]	a[4]
a	10	20	30	40	50

2.2.3 ถ้าในตอนประกาศตัวแปรอาร์เรย์ไม่กำหนดค่าเริ่มต้น ค่าที่อยู่ในตัวแปรจะเป็นค่าที่ ค้างอยู่ในหน่วยความจำช่วงที่เราจองไว้เป็นอาร์เรย์นั้น

2.2.4 ถ้ากำหนดค่าเริ่มต้นตั้งแต่ตอนประกาศตัวแปรแต่กำหนดไม่ครบ ในกรณีที่เป็นอาร์เรย์ แบบตัวเลขทั้งจำนวนเต็มและจำนวนจริง ค่าที่เหลือจะถูกกำหนดเป็น 0 โดยอัตโนมัติ

เช่น `float score[5] = {50.5, 2.25, 15.0};`

	score[0]	score[1]	score[2]	score[3]	score[4]
score	50.5	2.25	15.0	0.0	0.0
	4 bytes				

2.2.5 ถ้ากำหนดค่าให้แก่อาร์เรย์ตั้งแต่แรกไม่จำเป็นต้องใส่ขนาดของอาร์เรย์ก็ได้

เช่น `float a[ ] = {1, 2, 3, 4, 5};` เป็นการกำหนดตัวแปรอาร์เรย์ของจำนวนจริงแบบ float ขนาด 5 ช่อง

2.2.6 แต่ถ้ากำหนดตัวแปรอาร์เรย์โดยไม่ใส่ขนาดของอาร์เรย์และไม่ได้กำหนดค่าเริ่มต้น

เช่น `int a[ ] ;` ← **ประกาศผิด!!!**

นั่นคือ ไม่สามารถประกาศตัวแปรอาร์เรย์โดยไม่ใส่ขนาดของอาร์เรย์ได้ ยกเว้นมีการกำหนด ค่าเริ่มต้นให้ตั้งแต่แรก

2.3 การอ้างถึงข้อมูลย่อยแต่ละตัว ใช้เลขจำนวนเต็มทำหน้าที่เป็นดัชนี (Index) เพื่อระบุถึงข้อมูล นั้นโดยระบุชื่อตัวแปรอาร์เรย์แล้วตามด้วยค่าตำแหน่งที่อยู่ของสมาชิก ในภาษาซีดัชนีเริ่มตั้งแต่ [0] จนถึง [n-1] (n = จำนวนของสมาชิกในอาร์เรย์)

ตัวอย่างที่ 1 ให้ a เป็นอาร์เรย์ของ int ขนาด 5 เช่น โดยแต่ละช่องมีข้อมูล 1, 2, 3, 4, 5 ตามลำดับ

	a[0]	a[1]	a[2]	a[3]	a[4]
a	1	2	3	4	5

เขียนได้เป็น `int a[5];`



แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

หน่วยที่ 9

ชื่อหน่วย อาร์เรย์

สอนครั้งที่ 15

ชั่วโมงรวม 72

จำนวนชั่วโมง 4

```

a[0] = 1;
a[1] = 2;
a[2] = 3;
a [3] = 4;
a [4] = 5;

```

### ข้อสังเกต

1) a[0], a[4] เปรียบเสมือนตัวแปร int ธรรมดา 5 ตัว นั่นคือสามารถนำมากระทำการต่างๆ ได้ เหมือนตัวแปรธรรมดา เช่น รับข้อมูลเข้าไปเก็บไว้ในอาร์เรย์ a ตำแหน่งที่ 3 สามารถเขียนเป็นคำสั่งได้ ดังนี้

```

scanf ("%d", &a[3]);
i = 3;
a[i] = a[i+1] + a[4] ;

```

จะหมายถึง

- กำหนดค่าให้ตัวแปร i เท่ากับ 3
- นำข้อมูลในอาร์เรย์ a ตำแหน่งที่ i+1 มาบวกกับข้อมูลในอาร์เรย์ a ตำแหน่งที่ 4 แล้วเก็บ ผลลัพธ์ที่ได้ไว้ในอาร์เรย์ a ตำแหน่งที่ i (นั่นคือ นำ a[4] + a[4] แล้วเก็บผลลัพธ์นี้ไว้ใน a[3])

2) ขนาดของอาร์เรย์จะคงที่ เช่น กำหนดให้อาร์เรย์ N เก็บเลขจำนวนเต็มโดยมีขนาด 50 เช่น ก็หมายความว่า อาร์เรย์ N จะไปจองเนื้อที่ในหน่วยความจำขนาดที่เก็บจำนวนเต็มได้ 50 จำนวน โดยที่ จะเรียงกันไป 50 เช่นติดต่อกันและจะเก็บค่ามากกว่า 50 เช่นไม่ได้

ตัวอย่างที่ 2 โปรแกรมอาร์เรย์ 1 มิติ

```

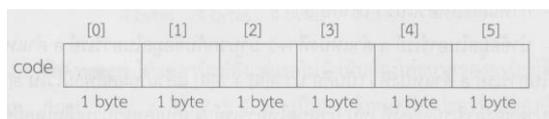
1  #include <stdio.h>
2  int main()
3  {
4      int n[5] = {5, 3, 2, 6, 1}, i, j;
5      for (i = 0; i < 5; i++)
6      {
7          for (j = 1; j <= n[i]; j++)
8              {printf("%*");}
9          printf("\n");
10     }
11     return 0;

```

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 9</b>
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่ 15
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4

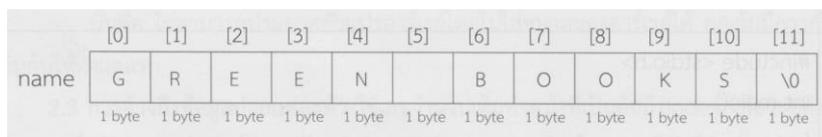
**2.4 สตริง (String) หรืออาร์เรย์ 1 มิติ ชนิดอักขระ** สามารถรวมอักขระแต่ละตัวเข้าด้วยกันเป็น สตริง (String) หรือข้อความ โดยประกาศให้ตัวแปรที่จะเก็บสตริงหรือข้อความเป็นตัวแปรอาร์เรย์ 1 มิติ ชนิดอักขระ

เช่น `char code[6];`



การประกาศให้ตัวแปร `code` เป็นอาร์เรย์ 1 มิติ ขนาด 6 เช่น แต่ละเซ่นเก็บข้อมูลประเภทอักขระ (ตัวอักษร 1 ตัว)

การกำหนดค่าเริ่มต้นให้กับสตริง `char name[ ] = "GREEN BOOKS";` จะได้



โดยที่ตำแหน่ง `name[11]` จะเก็บค่า `\0` ไว้อัตโนมัติ เพื่อแสดงการสิ้นสุดข้อความ หรือ `char name[13] = "GREEN BOOKS";` จะได้



12 }  
}

ผลการรันโปรแกรม :

```
*****
***
**
*****
```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

โดยที่ตำแหน่ง name[11] จะเก็บค่า \0 ไว้อัตโนมัติ เพื่อแสดงการสิ้นสุดข้อความเช่นเดียวกันและ ตำแหน่งที่ 12 จะว่าง

ดังนั้น ในการประกาศให้ตัวแปรเป็นสตริง จะต้องคำนึงถึงจำนวนตัวอักษรที่ต้องการจัดเก็บด้วย เช่น หากต้องการเก็บชื่อที่มีความยาวไม่เกิน 20 ตัวอักษร จะต้องประกาศตัวแปรให้เป็นอาร์เรย์ขนาด 21 เช่น (สำหรับเก็บค่า \0 หรือค่า null) นั่นคือ char name[21];

### 3. อาร์เรย์ 2 มิติ

โครงสร้างอาร์เรย์ 2 มิติ จะเป็นอาร์เรย์ของอาร์เรย์ 1 มิติ โดยสามารถอธิบายได้ในรูปแบบของตารางที่ประกอบด้วยแถว (Row) และคอลัมน์ (Column)

#### 3.1 รูปแบบของอาร์เรย์ 2 มิติ มีดังนี้

ชนิดข้อมูล ชื่อตัวแปร [row] [column];

เช่น int b[3][4];

		Column (คอลัมน์)			
		[0]	[1]	[2]	[3]
Row (แถว)	[0]	b[0][0]	b[0][1]	b[0][2]	b[0][3]
	[1]	b[1][0]	b[1][1]	b[1][2]	b[1][3]
	[2]	b[2][0]	b[2][1]	b[2][2]	b[2][3]

#### 3.2 การกำหนดค่าเริ่มต้นให้กับอาร์เรย์ 2 มิติ มีดังนี้

ตัวอย่างรูปแบบที่ 1 int num[2][3] = {11, 12, 13, 21, 22, 23};

ตัวอย่างรูปแบบที่ 2 int ทนทท[2][3] = {{11, 12, 13}, {21, 22, 23}};

ตัวอย่างรูปแบบที่ 3 int num[ ][3] = {{11, 12, 13}, {21, 22, 23}};

โดยที่การประกาศตัวแปรทั้ง 3 รูปแบบให้ผลลัพธ์เหมือนกัน คือ

num	[0]	[1]	[2]
[0]	11	12	13
[1]	21	22	23

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

### ข้อสังเกต

อาร์เรย์ขนาด 2 มิติขึ้นไป จะไม่ระบุขนาดได้เฉพาะมิติที่ 1 เท่านั้น ส่วนมิติอื่นๆ ต้องมีการระบุ ขนาดด้วยทุกครั้ง

### ตัวอย่างที่ 3 โปรแกรมอาร์เรย์ 2 มิติ

```

1 #include <stdio.h>
2 int main()
3 {
4     int i, j ;
5     int num[3][3] = {{ 1, 2, 3 } , { 4, 5, 6 }, { 7, 8, 9 }};
6     for (j = 2 ; j >= 0; j--)
7     {
8         for (i = 0; i <= 2; i++)
9             printf("%d\t", num[i][j]);
10        printf("\n");
11    }
12    return 0;
13 }
```

### โปรแกรม :

```

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

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

#### 4. อาร์เรย์หลายมิติ

สมมติประกาศให้  $k$  เป็นตัวแปรอาร์เรย์ 2 มิติ ขนาด  $3 \times 2$  (หมายถึง 3 แถวและ 2 คอลัมน์) เก็บข้อมูลประเภทจำนวนเต็ม

ดังนั้น จะได้ `int k[3][2];` นั่นคือ

k	[0]	[1]
[0]		
[1]		
[2]		

ซึ่งตัวแปร  $k$  ที่จัดเก็บอยู่ในหน่วยความจำหลักจริงๆ นั้นจะเรียงติดต่อกันไปเป็นแถวยาวไม่ได้ จัดเก็บอยู่ในรูปแบบตารางจริงๆ ดังนี้

k[0][0]	k[0][1]	k[1][0]	k[1][1]	k[2][0]	k[2][1]
แถวที่ 0		แถวที่ 1		แถวที่ 2	

ดังนั้น เราสามารถสร้างอาร์เรย์ซึ่งมีหลายมิติได้ไม่ว่าจะเป็นอาร์เรย์ 3 มิติ 4 มิติ หรือ 5 มิติ แต่ทั้งนี้ การที่ประกาศอาร์เรย์ที่มีจำนวนมิติมากๆ ก็จะทำให้โปรแกรมมีความซับซ้อนมากยิ่งขึ้น ส่งผลให้ การแก้ไขต่างๆ เป็นไปได้อย่างลำบาก

#### ตัวอย่างที่ 4 การประกาศอาร์เรย์ 3 มิติ

`int g[4][2][3];` คือ ประกาศอาร์เรย์  $g$  เก็บจำนวนเต็ม มี 4 แถว แถวละ 2 คอลัมน์ และ 3 ชั้น (1 ชั้น จะมี 4 แถว)

แถว 0	แถว 1	แถว 2	แถว 3	แถว 0	แถว 1	แถว 2	แถว 3	แถว 0	แถว 1	แถว 2	แถว 3			
ชั้น 0				ชั้น 1				ชั้น 2						

การอ้างถึงข้อมูลย่อยแต่ละตัวจะคล้ายกับอาร์เรย์ 1 มิติและ 2 มิติ นั่นคือ ใช้ดัชนีในการอ้างถึงโดยในที่นี้จะมีทั้งสิ้น 3 ตัว คือ แถว คอลัมน์ และชั้น

เช่น `g[0][1][2] = 50;` หมายถึง ให้เก็บ 50 ไว้ในอาร์เรย์  $g$  ตำแหน่งแถวที่ 0 คอลัมน์ที่ 1 และชั้นที่ 2

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 9</b>
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่ 15
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4

## 5. การเรียงลำดับ

การทำงานโดยส่วนใหญ่จำเป็นต้องมีการเรียงลำดับข้อมูลก่อนการเก็บข้อมูล เพื่อสะดวกในการค้นหา การเรียงลำดับมีหลายวิธี เช่น การเรียงลำดับแบบฟอง (Bubble Sort) การเรียงลำดับแบบเลือก (Selection Sort) การเรียงลำดับแบบแทรกตำแหน่ง (Insertion Sort) การเรียงลำดับแบบเร็ว (Quick Sort) การเรียงลำดับเชิงสถิติ (Order Statistics) การเรียงลำดับโดยใช้ฮีป (Heap Sort) การเรียงลำดับ โดยวิธีของ Shell (Shell Sort) การเรียงลำดับโดยอาศัยฐานตัวเลข (Radix Sort) และอื่นๆ อีกมากมาย แต่ในที่นี้จะนำมาศึกษา 3 วิธี คือ การเรียงลำดับแบบฟอง (Bubble Sort) การเรียงลำดับแบบเลือก (Selection Sort) และการเรียงลำดับแบบแทรกตำแหน่ง (Insertion Sort) เพื่อใช้ในการศึกษาวิธีการ เรียงลำดับ

**5.1 การเรียงลำดับแบบฟอง (Bubble Sort)** เป็นการเรียงลำดับที่มีการตรวจสอบค่าทีละค่า แล้วสลับค่าให้ถูกต้อง การเรียงลำดับจากน้อยไปมากด้วยวิธี Bubble Sort จะมีหลักการเรียงลำดับดังนี้ ถ้าตัวปัจจุบันมีค่ามากกว่าตัวถัดไปจะสลับที่กันและเลื่อนตำแหน่งขึ้น ในกรณีต้องการเรียงจากมากไปน้อย ถ้าตัวปัจจุบันมีค่าน้อยกว่าตัวถัดไปจะสลับที่กันและเปลี่ยนตำแหน่งปัจจุบันเป็นตำแหน่งถัดไปเรื่อยๆ จนกว่า จะครบทุกตัว กระทำเช่นนี้ไปเรื่อยๆ จนกว่าชุดข้อมูลนั้นจะเรียงเรียบร้อย

### รูปแบบอัลกอริทึมการเรียงลำดับแบบฟอง (Bubble Sort)

1	procedure BubbleSort(A : list of sortable items) defined as:
2	for each i in 1 to length(A) do:
3	for each j in length(A) downto i + 1 do:
4	if $A[j - 1] > A[j]$ then
5	swap( $A[j - 1]$ , $A[j]$ )
6	end if
7	end for
8	end for
9	end procedure

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 9</b>
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่ 15
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4

วิธีการเรียงลำดับแบบฟองมีการเขียนโปรแกรมลูป 2 ลูป คือ ลูป i รันจำนวนรอบตามจำนวน ของข้อมูล และ ลูป j เป็นลูปที่เปรียบเทียบค่าแต่ละตัวว่าเรียงลำดับถูกต้องหรือไม่ หากค่ามากอยู่ก่อน ค่าน้อยก็จะสลับค่าเมื่อสลับค่าในแต่ละรอบจะได้ค่ามากที่สุดอยู่ท้ายสุดดังนั้น หากต้องการเขียนโปรแกรม ภาษาซีเพื่อเรียงลำดับแบบฟอง (Bubble Sort) สามารถทำได้ดังโปรแกรมตัวอย่างที่ 5

ตัวอย่างที่ 5 การเรียงลำดับแบบฟอง (Bubble Sort)

```

1  #include <stdio.h>
2  int main()
3  {
4      int array[100], n, c, d, swap;
5      printf("Enter number of elements\n");
6      scanf("%d", &n);
7      printf("Enter %d integers\n", n);
8      for (c = 0; c < n; C++)
9          scanf("%d", &array[c]);
10     for (c = 0; c < n-1; C++)
11     {
12         for (d = 0; d < n-c-1; d++)
13         {
14             if (array[d] > array[d+1]) /* For decreasing order use < */
15             {
16                 swap = array[d];
17                 array[d] = array[d+1];
18                 array[d+1] = swap;
19             }
20         }
21     }
22     printf("Sorted list in ascending order:\n");
23     for (c = 0; c < n; C++)
24         printf("%d\n", array[c]);
25     return 0;
26 }
```

ผลการรันโปรแกรม :

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<div style="border: 1px solid black; padding: 10px;"> <p>Enter number of elements</p> <p>6</p> <p>Enter 6 integers</p> <p>9</p> <p>2</p> <p>8</p> <p>4</p> <p>6</p> <p>6</p> <p>Sorted list in ascending order:</p> <p>2</p> <p>4</p> <p>6</p> <p>6</p> <p>8</p> <p>9</p> </div>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>9</b>
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูชักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	9
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่	15
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>10</b>
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>1. สาระสำคัญ</b></p> <p>พอยน์เตอร์ เป็นตัวแปรที่ใช้เก็บค่าแ่งที่อยู่ของตัวแปร การใช้พอยน์เตอร์ เราสามารถเข้าถึงข้อมูลที่เราต้องการได้โดยตรงโดยการใช้ชื่อของตัวแปร และมันมีประโยชน์มากในการเขียนโปรแกรมด้วยภาษาที่สามารถจัดการหน่วยความจำได้ง่าย เช่น ภาษา C และภาษา C++</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับพอยน์เตอร์ได้</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรมภาษาอื่น ๆ</p> <p>3.1.2 เขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

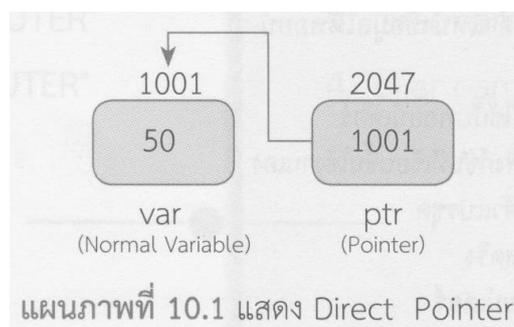
#### 4. เนื้อหาสาระการเรียนรู้

##### 1. ความหมายและประเภทของพอยน์เตอร์

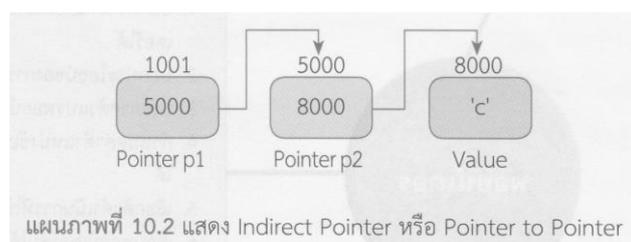
1.1. ความหมายของพอยน์เตอร์ พอยน์เตอร์คือ ตัวแปรชนิดหนึ่งที่เก็บตำแหน่ง (Address) ของข้อมูลภายในหน่วยความจำ ซึ่งการเก็บตำแหน่งจะเก็บเฉพาะตำแหน่งแรกๆของข้อมูลเท่านั้น

1.2 ประเภทของพอยน์เตอร์ สามารถแบ่งออกเป็น 2 ชนิด ดังนี้

1.2.1 Direct Pointer คือ ตัวแปรที่เก็บตำแหน่งของข้อมูลภายในหน่วยความจำโดยตรง



1.2.2 Indirect Pointer คือ ตัวแปรที่เก็บตำแหน่งของพอยน์เตอร์อีกตัวหนึ่ง บางครั้งอาจเรียกว่า Pointer to Pointer ก็ได้



##### 2. ประโยชน์ของการใช้พอยน์เตอร์

การนำพอยน์เตอร์มาใช้งานในโปรแกรมขึ้นอยู่กับวัตถุประสงค์ของการใช้งานโดยส่วนมากนิยมนำพอยน์เตอร์มาช่วยงาน ดังนี้

ใช้รับค่า Address ของตัวแปรชุด (Arrays) หรือตัวแปรสตริง (Strings Variables) จากฟังก์ชันหนึ่งไปยังอีกฟังก์ชันหนึ่ง และใช้ในการส่งค่ากลับมายังฟังก์ชัน

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p>2.1 ใช้ในการจัดการตัวแปรชุดหรือตัวแปรสตริงให้มีประสิทธิภาพ โดยการอ้างอิงตำแหน่งของ ตัวแปรชุดหรือตัวแปรสตริงที่ต้องการแทนที่การอ้างอิงชื่อตัวแปรชุดหรือตัวแปรสตริงโดยตรง ทำให้สามารถจัดเก็บและเรียกใช้ข้อมูลที่ต้องการได้ง่าย</p> <p>2.2 ใช้พอยน์เตอร์ช่วยในการจัดการโครงสร้างข้อมูล เช่น ลิงก์ลิสต์ (Linked Lists), ไบนารีทรี (Binary Tree), การจัดเรียงข้อมูล (Sort) ชนิดต่างๆ เป็นต้น</p>			
<h3>3. การประกาศตัวแปรพอยน์เตอร์</h3> <p>ก่อนที่จะนำตัวแปรพอยน์เตอร์ไปใช้งาน ต้องมีการประกาศตัวแปรพอยน์เตอร์ ดังนี้</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <pre> type *ptr_name;           กรณีที่พอยน์เตอร์ 1 ตัว หรือ type *prt1_name, *prt2_name, *prt3_name, ...  กรณีที่พอยน์เตอร์มากกว่า 1 ตัว </pre> </div> <p>type คือ ชนิดของตัวแปรพอยน์เตอร์ซึ่งจะต้องเป็นชนิดเดียวกับข้อมูล นั่นหมายความว่า ถ้าข้อมูล ที่เก็บไว้ตรงตำแหน่งที่ตัวแปรพอยน์เตอร์ชื่ออยู่เป็นข้อมูลชนิด int ตัวแปรพอยน์เตอร์จะต้องมีชนิด int ด้วย สำหรับขนาดของตัวแปรพอยน์เตอร์ก็ขึ้นอยู่กับชนิดของตัวแปรพอยน์เตอร์</p> <p>* คือ เครื่องหมาย Asterick ที่แสดงให้ทราบว่าตัวแปรที่ตามหลังเครื่องหมายนี้เป็นตัวแปรพอยน์เตอร์</p> <p>ptr_name คือ ชื่อตัวแปรพอยน์เตอร์โดยจะต้องมีเครื่องหมาย * (Asterisk) นำหน้าชื่อเพื่อบอกให้ compiler รู้ว่าเป็นตัวแปรพอยน์เตอร์</p> <p>prt1_name, prt2_name, ... คือ ชื่อตัวแปรพอยน์เตอร์ตัวที่ 1, 2,... ตามลำดับ (กรณีมากกว่า 1 ตัว)</p> <p><b>ตัวอย่างที่ 1</b> แสดงการประกาศตัวแปรพอยน์เตอร์</p> <pre> int *ptr; float *pone, *ptwo, *pthree; char *ptrx, *ptry, *ptrz; </pre>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

#### 4. การกำหนดค่าตำแหน่งข้อมูลให้พอยน์เตอร์

เมื่อต้องการกำหนดค่า Address แรกของข้อมูลให้กับตัวแปรพอยน์เตอร์ ในภาษาซีสามารถใช้รูปแบบได้ดังนี้

```
ptr_name = &variable_name
```

โดยที่

ptr\_name คือ ชื่อตัวแปรพอยน์เตอร์

variable\_name คือ ชื่อตัวแปรที่ต้องการกำหนดค่า Address แรกให้ตัวแปรพอยน์เตอร์ในกรณีที่เป็นตัวแปรชุด (Arrays Variables) หรือตัวแปรสตริง (String Variables) ไม่ต้องใส่เครื่องหมาย & (Ampersand) นำหน้า เช่น char s1[80] = "computer"; char \*ptrs1; ptrs1 = s1;

**ตัวอย่างที่ 2** การกำหนด Address ให้กับตัวแปรพอยน์เตอร์

```
int p, q, *pp, *qq; /* ประกาศตัวแปร p, q และตัวแปรพอยน์เตอร์ PP และ qq */
pp = &p /* กำหนดค่า Address แรกของตัวแปร P ไปเก็บไว้ที่พอยน์เตอร์ PP */
qq = &q /* กำหนดค่า Address แรกของตัวแปร q ไปเก็บไว้ที่พอยน์เตอร์ qq */
```

**ตัวอย่างที่ 3** การใช้ตัวดำเนินการ \* เพื่อบอกถึงค่าข้อมูลที่พอยน์เตอร์ชี้อยู่

```
1 #include <stdio.h>
2 int main (void)
3 {
4     int a = 10, b = 25;
5     int *pa, *pb;
6     pa = &a;
7     pb = &b;
8     Printf(a = %d, b = %d", *pa, *pb);
9     return 0;
10 }
```

ผลการรันโปรแกรม :

a = 10, b = 25

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

### คำอธิบายโปรแกรม

จากโปรแกรมตัวอย่างที่ 3 สามารถอธิบายการทำงานของโปรแกรมที่สำคัญๆ ได้ดังนี้  
 บรรทัดที่ 5 คำสั่ง `int *pa, *pb;` เป็นการประกาศตัวแปรพอยน์เตอร์ `pa` และ `pb` เป็นชนิด `int`  
 บรรทัดที่ 6 คำสั่ง `pa = &a;` เป็นการกำหนดค่า Address แรกของ `a` ให้ตัวแปรพอยน์เตอร์ `pa`  
 บรรทัดที่ 7 คำสั่ง `pb = &b;` เป็นการกำหนดค่า Address แรกของ `b` ให้ตัวแปรพอยน์เตอร์ `pb`  
 บรรทัดที่ 8 พิมพ์ค่าที่ตัวแปรพอยน์เตอร์ `pa` และ `pb` ซ้ำๆ แล้วแสดงออกทางจอภาพ นั่นคือ

ค่าตัวแปร `a` และ `b` ตามลำดับ

### 5. ตัวดำเนินการที่ใช้กันพอยน์เตอร์

ตัวดำเนินการ `*` (Asterisk) บางครั้งเรียกว่า เครื่องหมายดอกจัน เป็นเครื่องหมายที่ใช้บ่งบอกถึงค่าข้อมูลที่พอยน์เตอร์ชี้อยู่เพื่อความเข้าใจเกี่ยวกับการใช้ตัวดำเนินการ `*` กับพอยน์เตอร์มากยิ่งขึ้น จะแสดงดังตัวอย่างที่ 4

ตัวอย่างที่ 4 การใช้ตัวดำเนินการ `&` เพื่อบอกถึงค่า Address ของข้อมูล

```

1 #include <stdio.h>
2 void main(void)
3 {
4     int m = 25, n = 60;
5     int *pm1 *pn;
6     pm = &m;
7     pn = &n;
8     printf("M1 = %d, N1 = %d\n", m, n);
9     printf("M2 = %d, N2 = %d\n", *pm, *pn);
10    printf("\nAddress of m, n are %p %p \n", &pm, &pn);
11    getch();
12}

```

ผลการรันโปรแกรม :

M1 = 25, N1 = 60

M2 = 25, N2 = 60

Address of m, n are 0023FF14 0023FF10

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>คำอธิบายโปรแกรม</b></p> <p>บรรทัดที่ 5 คำสั่ง <code>int *pm,*pn</code>; คำสั่งประกาศตัวแปรพอยน์เตอร์ <code>pm</code> และ <code>pn</code> เป็นชนิด <code>int</code></p> <p>บรรทัดที่ 6 คำสั่ง <code>pm = &amp;m</code>; เป็นการกำหนดค่า Address แรกของ <code>ทก</code> ให้ตัวแปรพอยน์เตอร์ <code>pm</code></p> <p>บรรทัดที่ 7 คำสั่ง <code>pn = &amp;n</code>; เป็นการกำหนดค่า Address แรกของ <code>n</code> ให้ตัวแปรพอยน์เตอร์ <code>pn</code></p> <p>บรรทัดที่ 8 พิมพ์ค่าตัวแปร <code>m</code> และ <code>n</code> แสดงที่จอภาพ</p> <p>บรรทัดที่ 9 พิมพ์ค่าตัวแปรที่ตัวแปรพอยน์เตอร์ <code>pm</code> และ <code>pn</code> ซี่อยู่แสดงที่จอภาพ</p> <p>บรรทัดที่ 10 พิมพ์ค่า Address ของตัวแปร <code>m</code> และ <code>n</code> ที่ตัวแปรพอยน์เตอร์ <code>pm</code> และ <code>pn</code> ซี่อยู่ โดยใช้รหัสรูปแบบข้อมูลเป็น <code>%p</code> แสดงที่จอภาพ</p> <p><b>6. พอยมตอ์กับฟังก์ชันที่เขียนส้นใจงานเอง</b></p> <p>สามารถใช้ตัวแปรพอยน์เตอร์มาช่วยในการรับค่าและส่งคืนค่ากลับในฟังก์ชันที่เขียนขึ้นได้ดังนี้</p> <p><b>6.1 การส่งค่าไปยังฟังก์ชัน(Passing Address to Function) สามารถทำได้ 2 วิธี</b></p> <p><b>6.1.1การส่งค่าข้อมูลไปยังฟังก์ชันที่ต้องการ (Passing Data Values to Function)</b></p> <p>การส่งค่าข้อมูล 1 ค่าหรือหลายค่าจากฟังก์ชันหนึ่งไปอีกฟังก์ชันหนึ่ง คือ การคัดลอกค่าข้อมูล (Duplicated Value) จากที่หนึ่งไปยังอีกที่หนึ่ง</p> <p><b>6.1.2 การส่งตำแหน่งของข้อมูลไปยังฟังก์ชัน (Passing Address to Function)</b></p> <p>การส่งตำแหน่งของข้อมูลไปยังฟังก์ชันที่ต้องการ จะต้องใช้ตัวแปรพอยน์เตอร์มาช่วยในการเก็บตำแหน่งของข้อมูลที่ส่งมา เพื่อความเข้าใจเกี่ยวกับการส่งค่าข้อมูลและตำแหน่งของข้อมูลไปยังฟังก์ชันที่ต้องการ</p> <p><b>ตัวอย่างที่ 6 การส่งค่าข้อมูลและตำแหน่งของข้อมูลไปยังฟังก์ชันที่ต้องการ</b></p> <pre> 1 #include&lt;stdio.h&gt; 2 void f1(int, int);      /* function prototype f1 */ 3 void f2(int*, int*);   /* function prototype f2 */ 4 void main(void) 5 { 6     int P = 15, q = 30; 7     f1(p,q);           /* called f1() */ 8     f2(&amp;p,&amp;q);         /* called f2() */ 9     return 0; 10 } /* end main */ </pre>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

ตัวอย่างที่ 6 การส่งค่าข้อมูลและตำแหน่งของข้อมูลไปยังฟังก์ชันที่ต้องการ (ต่อ)

```

11 void fl(int pp, int qq) {
12     Printf('P1 = %d\tQ1 = %d\n", pp, qq);
13 }
14 void f2(int *pp, int *qq) {
15     *pp = *pp + 20;
16     *qq = *qq + 20;
17     printf("P2 = %d\tQ2 = %d\n", *pp, *qq);
18 }

```

P1 = 15 Q1 = 30

P2 = 35 Q2 = 50

ผลการรันโปรแกรม :

คำอธิบายโปรแกรม

บรรทัดที่ 2 คำสั่ง void fl(int, int); เป็นการประกาศรูปแบบฟังก์ชันf1

บรรทัดที่ 3 คำสั่ง void f2(int\*, int\*); เป็นการประกาศรูปแบบฟังก์ชันf2

บรรทัดที่ 7 คำสั่ง fl(p,q); การเรียกใช้งานฟังก์ชัน fl และส่งข้อมูลของตัวแปร P กับ q ไปยัง ฟังก์ชัน fl ซึ่งฟังก์ชันf1 เป็นคำสั่งที่อยู่ในบรรทัดที่ 11 ถึง 13

บรรทัดที่ 8 คำสั่ง f2(&p,&q); การเรียกใช้งานฟังก์ชันf2 และส่งข้อมูลของตัวแปร P กับ q ไปยัง ฟังก์ชันf2 ซึ่งฟังก์ชันf2 เป็นคำสั่งที่อยู่ในบรรทัดที่ 14 ถึง 18

บรรทัดที่ 7 ถึง 8 เป็นฟังก์ชันที่ตัวแปรพอยน์เตอร์ pp, qq จะรับค่าข้อมูลจากตัวแปร P และ q ที่ส่งมาให้จากจุดที่เรียกใช้งานในฟังก์ชันmain ตามลำดับ และแสดงค่าตัวแปรที่ตัวแปรพอยน์เตอร์ PP และ qq รับค่ามาจากตัวแปร P และ q ออกที่จอภาพ

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 10</b>
	ชื่อหน่วย อาร์เรย์	สอนครั้งที่ 16
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4
<p>บรรทัดที่ 11 ถึง 18 เป็นฟังก์ชันที่ตัวแปรพอยน์เตอร์ pp, qq จะรับค่าตำแหน่งของตัวแปร P และ q ที่ส่งมาให้จากจุดที่เรียกใช้งานในฟังก์ชัน main ตามลำดับ เมื่อรับค่าแล้วให้บวกเพิ่มค่าละ 20 หลังจากนั้นนำค่าที่ตัวแปรพอยน์เตอร์ PP และ qq ซึ่อยู่แสดงออกจอภาพ</p> <p><b>6.2 การคืนค่ากลับของฟังก์ชัน (Returning Data from Function)</b> โดยปกติการคืนค่า กลับมายังฟังก์ชันสามารถทำได้เพียง 1 ค่าเท่านั้น แต่ในกรณีที่ต้องการให้มีการคืนค่ากลับหลายค่าต้องใช้ พอยน์เตอร์ มาช่วยจึงสามารถกระทำได้ โดยมีขั้นตอนดังนี้</p> <p><b>ขั้นตอนแรก</b> ในโปรแกรมที่ใช้เรียกฟังก์ชัน (Calling Program) จะต้องใช้วิธีการส่งตำแหน่ง ของข้อมูลไปยังฟังก์ชัน โดยใช้เครื่องหมาย &amp; นำหน้าชื่อตัวแปรที่ต้องการส่งค่า Address ไป</p> <p><b>ขั้นตอนที่สอง</b> ในฟังก์ชันที่ถูกเรียกใช้ จะต้องใช้ตัวแปรพอยน์เตอร์มารับค่า Address ที่ ส่งมาเมื่อต้องการคืนค่ากลับให้ใช้เครื่องหมาย * นำหน้าชื่อตัวแปรพอยน์เตอร์ เพื่อคืนค่ากลับมายังตัวแปร ของฟังก์ชัน</p> <p><b>ตัวอย่างที่ 7</b> การคืนค่ากลับของฟังก์ชันที่มากกว่า 1 ค่า</p> <pre> 1 #include &lt;stdio.h&gt; 2 void back(int *, int *); /* function prototype backO V 3 int main() 4 { 5     int a, b; 6     back(&amp;a,&amp;b); /* called backO V 7     Printf('A = %d, B = %d\n", a, b); 8     return 0; 9 } /* end main */ 10 void back(int *aa, int *bb) 11 { 12     *aa = 200; /* assigned 300 pass aa to a V 13     *bb = 400; /* assigned 400 pass bb to b V 14 } 14 } </pre> <p><b>ผลการรันโปรแกรม :</b></p> <p style="text-align: center;">A = 200, B = 400</p>		

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

### อธิบายโปรแกรม

บรรทัดที่ 2 คำสั่ง `void back(int *, int *);` เป็นการประกาศรูปแบบฟังก์ชัน `back`

บรรทัดที่ 6 ในฟังก์ชัน `main()` มีการเรียกใช้ฟังก์ชัน `back(&a,&b);` หมายถึง มีการส่งค่า

Address ของตัวแปร `a` และ `b` ไปยังตัวแปรพอยน์เตอร์ `aa` และ `bb` ของฟังก์ชัน `back` ตามลำดับ ซึ่งฟังก์ชัน `back` อยู่คำสั่งบรรทัดที่ 10 ถึง 14 ซึ่งมีการใช้เครื่องหมาย `*` อยู่หน้าตัวแปรพอยน์เตอร์ `aa` และ `bb` เพื่อบอกให้กำหนดค่าตัวเลข 200 และ 400 ให้กับตัวแปร `a` และ `b` เนื่องจากภายในตัวแปรพอยน์เตอร์ ทั้งสองตัวได้เก็บค่า Address แรกของตัวแปร `a` และ `b` ไว้แล้ว จึงเหมือนกับการส่งค่ากลับ 2 ค่า พร้อมกัน

### 7. พอยน์เตอร์กับตัวแปรชุด

พอยน์เตอร์สามารถนำมาช่วยในการจัดการข้อมูลที่เกิดขึ้นในตัวแปรชุดได้ ซึ่งจะช่วยให้ การจัดเก็บและการเรียกใช้ข้อมูลในตัวแปรชุดมีประสิทธิภาพมากขึ้น เพื่อความเข้าใจเกี่ยวกับการใช้งาน พอยน์เตอร์กับตัวแปรชุด ให้ศึกษาโปรแกรมในตัวอย่างที่ 8

#### ตัวอย่างที่ 8 การใช้พอยน์เตอร์กับตัวแปรชุด

```

1 #include <stdio.h>
2 int main ()
3 {
4     int num[10] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
5     int *ptr; /* declaration pointer ptr */
6     int i;
7     ptr = &num[0];
8     for (i = 0; i <= 9; i++)
9         printf("%5d", *(ptr+i));
10 return 0;
11 }
```

#### ผลการรันโปรแกรม :

```

10  20  30  40  50  60  70  80  90  100
```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

### คำอธิบายโปรแกรม

บรรทัดที่ 5 คำสั่ง `int *ptr`; การประกาศตัวแปรพอยน์เตอร์ชื่อ `ptr` ชนิด `int`

บรรทัดที่ 7 คำสั่ง `ptr = &num[0]`; เป็นการกำหนด Address แรกของตัวแปร `num[0]` ให้กับ ตัวแปร `ptr` หรือใช้คำสั่ง `ptr = num`; ก็ได้

บรรทัดที่ 8 คำสั่ง `for` เพื่อช่วยวนลูปนำค่าที่ตัวแปรอาร์เรย์ชื่ออยู่ออกมาแสดง

บรรทัดที่ 9 คำสั่ง `printf("%5d", *(ptr+i))`; เป็นการพิมพ์ค่าในตัวแปรอาร์เรย์ `num` โดยใช้ ตัวแปรพอยน์เตอร์ ซึ่งนิพจน์ `*(ptr+i)` หมายถึง การพิมพ์ค่าที่ตัวแปรพอยน์เตอร์ชื่ออยู่ คือ `num[i]`

### 8. พอยน์เตอร์กับสตริง

เนื่องจากการเก็บค่าคงที่ชนิดสตริงจะเก็บไว้ในตัวแปรสตริง ดังนั้น เมื่อสามารถใช้พอยน์เตอร์กับ ตัวแปรชุดได้ ก็ย่อมสามารถใช้พอยน์เตอร์มาจัดการกับค่าคงที่ชนิดสตริงได้เช่นกัน

#### ตัวอย่างที่ 9 การใช้พอยน์เตอร์กับสตริงในการตั้งค่าสตริงและนำออกมาแสดงทางจอภาพ

```

1 #include<stdio.h>
2 int main()
3 {
4     char *ptr = "Hello";
5     char name[80];
6     printf("Enter your name : ");
7     gets(name);
8     printf("\n%s, %s", ptr, name);
9     return 0;
10 }
```

### ผลการรันโปรแกรม :

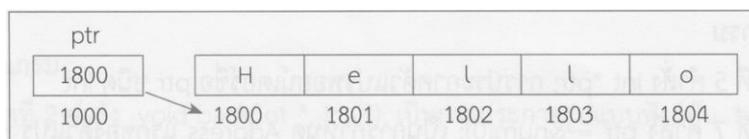
Enter your name : computer

Hello. computer

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

### คำอธิบายโปรแกรม

บรรทัดที่ 4 คำสั่ง `char *ptr = "Hello";` มีผลทำให้ภายในหน่วยความจำมีลักษณะดังนี้



บรรทัดที่ 7 ถึง 8 รับ

ค่าสตริงที่ผู้ใช้ป้อนเก็บไว้ที่ตัวแปร `name` แล้วนำค่าที่เก็บไว้ในตัวแปร พอยน์เตอร์ `ptr` ชี้อยู่ และตัวแปร `name` แสดงออกที่จอภาพ

### 9. อาร์เรย์ของพอยน์เตอร์

ในกรณีที่มีความต้องการใช้ตัวแปรพอยน์เตอร์เหมือนกับตัวแปรชุดก็สามารถกำหนดได้ตามรูปแบบดังนี้

```
type *ptrname[size];
หรือ
type *ptrname[size] = {value list};
```

`type` คือ ชนิด

ของข้อมูลซึ่งอาจจะ

เป็น `int`, `float`, `double` หรือ `char` ก็ได้ ในกรณีที่เป็น `char` ค่าที่เก็บจะต้องเป็นค่าคงที่ชนิดสตริง

`value list` คือ ค่าตัวเลขหรือค่าคงที่ชนิดสตริงขึ้นอยู่กับว่า `type` เป็น `int`, `float`, `double` หรือ `char` ถ้า `type` เป็น `char` จะทำให้ `value list` เป็นค่าคงที่ชนิดสตริง ซึ่งต้องเขียนอยู่ภายใต้เครื่องหมาย "" (Double Quotation)

`ptrname` คือ ชื่อตัวแปรพอยน์เตอร์ ต้องใช้เครื่องหมาย \* (Asterisk) นำหน้าชื่อเสมอ

`size` คือ ขนาดของตัวแปรชุดพอยน์เตอร์

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>10</b>
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและชวนชื่อผู้เรียน</p> <p>5.1.2 ครูชักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	10
	ชื่อหน่วย พอยน์เตอร์	สอนครั้งที่	16
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>11</b>
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>1. สาระสำคัญ</b></p> <p>การสร้างชนิดข้อมูล ก่อนจะศึกษาชนิดข้อมูลแบบ โครงสร้างและยูเนียนนั้น จะต้องรู้จักคำสั่งในการสร้างชนิดข้อมูลก่อน เพราะข้อมูลแบบโครงสร้างและยูเนียน จะใช้คำสั่งนี้สร้างขึ้นมา นั่นก็คือ typedef ผู้ใช้จะสามารถสร้างข้อมูลใหม่จากชนิดข้อมูลพื้นฐานที่มีอยู่เดิมแล้ว โดยรูปแบบคำสั่งของ typedef ดังนี้</p> <p>รูปแบบ Typedef [ชนิดข้อมูล] [ชื่อชนิดข้อมูลใหม่] ตัวอย่างข้างล่างนี้เป็นการสร้างและใช้ชนิดข้อมูลนั้นๆ</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับข้อมูลชนิดโครงสร้างและยูเนียน</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p>3.1. ด้านความรู้</p> <p>3.1.1 แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรมภาษาอื่น ๆ</p> <p>3.1.2 เขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p>3.2 ด้านทักษะ</p> <p>-</p> <p>3.3 คุณลักษณะที่พึงประสงค์</p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>11</b>
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

## 4. เนื้อหาสาระการเรียนรู้

### 1. นิยามและการประมวลผลสตรักเจอร์

สตรักเจอร์เป็นการรวมตัวแปรหลายๆ ตัวแปรเข้าด้วยกันให้อยู่ในชื่อเดียวกัน เพื่อสะดวกต่อการ เก็บข้อมูลเป็นหมวดหมู่ และเป็นประโยชน์ในการเขียนโปรแกรมให้ง่ายต่อการอ่านและทำความเข้าใจ การประกาศข้อมูลแบบโครงสร้างมี 2 รูปแบบ คือ ประกาศข้อมูลแบบโครงสร้างก่อนแล้วจึงกำหนดตัวแปร ให้มีข้อมูลเป็นแบบโครงสร้าง และประกาศข้อมูลแบบโครงสร้างพร้อมกับกำหนดตัวแปรพร้อมกัน แต่ทั้ง สองแบบใช้คำสั่งร่วมกัน คือ struct

**รูปแบบที่ 1** กำหนดโครงสร้างพร้อมกับการประกาศตัวแปรที่มีโครงสร้างนั้น มีรูปแบบดังนี้

```
struct structure name
{
    type 1 name 1;
    type 2 name 2;
    :
    type n name n;
}
structure variable 1, structure variable 2, ..., structure variable n;
```

**รูปแบบที่ 2** กำหนดโครงสร้างก่อนแล้วจึงประกาศตัวแปรที่มีโครงสร้างนั้นในภายหลัง มีรูปแบบดังนี้

```
struct structure name
{
    type 1 name 1;
    type 2 name 2;
    :
    type n name n;
};
struct structure name structure variable 1, structure variable 2, ..., structure variable n;
```

โดย

struct เป็นคำสั่งสำหรับประกาศข้อมูลแบบโครงสร้าง



แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

หน่วยที่ 11

ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน

สอนครั้งที่ 17

ชั่วโมงรวม 72

จำนวนชั่วโมง 4

structure name เป็นชื่อของข้อมูลแบบโครงสร้างใช้สำหรับกำหนดข้อมูลกลุ่มอื่นๆ ให้มีโครงสร้าง แบบเดียวกัน

type 1, type 2, type n คือ ชนิดของตัวแปรหรือข้อมูลต่างๆ เช่น int float char เป็นต้น name 1, name 2, name n คือ ชื่อของตัวแปรต่างๆ

structure variable เป็นชื่อของตัวแปรที่โครงสร้างเหมือนกับข้อมูลโครงสร้างชื่อ structure name

**ตัวอย่างที่ 1** รูปแบบที่ 1

```
struct person
{
    char name[20];
    char surname[20];
    double salary;
    int birthday;
} somchat, sunee, wimol;
```

**ตัวอย่างที่ 2** รูปแบบที่ 2

```
struct person
{
    char name[20];
    char surname[20];
    double salary;
    int birthday;
};
struct person somchat, sunee, wimol;
```

## 2. การอ้างถึงสมาชิกของตัวแปรโครงสร้าง

การกำหนดข้อมูลให้ตัวแปรโครงสร้างและการแสดงผลตัวแปรโครงสร้าง นำหลักการเข้าถึงตัวแปร ที่เป็นสมาชิกของตัวแปรมีรูปแบบ ดังนี้

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่ 11</b>
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่ 17
		ชั่วโมงรวม 72
		จำนวนชั่วโมง 4

### structure variable.name

โดย structure variable คือ ชื่อของตัวแปรโครงสร้าง

ส่วน name คือ ชื่อของตัวแปรที่เป็นสมาชิกในโครงสร้างนั้นโดยมีจุดคั่นระหว่างตัวแปรโครงสร้างกับตัวแปรสมาชิก

ส่วนการกำหนดข้อมูลให้แก่ตัวแปรโครงสร้างและการแสดงผลตัวแปรโครงสร้าง นำหลักการ เข้าถึงตัวแปรที่เป็นสมาชิกของตัวแปรโครงสร้าง การกำหนดข้อมูลให้แก่ตัวแปร และการแสดงผลตัวแปร ตามชนิดของตัวแปรมาใช้ประกอบกันในการระบุหรืออ้างอิงถึงตัวแปร ซึ่งเป็นสมาชิกในโครงสร้างของ ตัวแปรโครงสร้างนั้น เช่น ในตัวแปรโครงสร้าง phanee จะมีสมาชิกที่อ้างถึงได้ดังนี้

phanee.name

phanee.surname

phanee.salary

phanee.birthday

### 3. การประกาศโครงสร้างตัวแปรแบบอาร์เรย์

ถ้าต้องการเก็บข้อมูลแบบโครงสร้างจำนวนมากๆ สามารถกำหนดเป็นอาร์เรย์ของตัวแปรโครงสร้างได้ เช่น ต้องการเก็บข้อมูลโครงสร้างของนักศึกษา 20 คน ก็สามารถทำได้โดยการประกาศตัวแปรแบบอาร์เรย์ที่มีสมาชิก 20 ตัว โดยที่สมาชิกแต่ละตัวเป็นตัวแปรชนิดโครงสร้างของนักศึกษา โดยมีรูปแบบดังนี้

```
struct student
```

```
{
```

```
    int id; //รหัสประจำตัว
```

```
    char name[20]; //ชื่อ
```

```
};
```

```
struct student students[50]; //ประกาศตัวแปรอาร์เรย์ของโครงสร้างนักศึกษา 50 คน
```

ถ้าต้องการกำหนดค่ารหัสประจำตัวให้กับนักศึกษาคนที่ 20 และพิมพ์ออกมาทางหน้าจอ สามารถทำได้ดังนี้

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	11
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<pre>students[19].id = 1234; printf("%d", students[19].id);</pre>			
<p><b>4. การกำหนดโครงสร้างซ้อนโครงสร้าง</b></p> <p>การใช้งานตัวแปรแบบโครงสร้างซ้อนโครงสร้าง หมายถึง สมาชิกย่อยในโครงสร้างเป็นตัวแปร แบบโครงสร้าง เช่น ถ้าต้องการเก็บข้อมูลวัน เดือน ปีเกิดของนักศึกษา จะต้องสร้างโครงสร้างของวันเดือน ปี โดยจะให้ชื่อว่า date และ date เป็นสมาชิกของโครงสร้าง student ใช้ชื่อว่า birthdate หมายถึง วันเดือนปีเกิด โดยสามารถเขียนเป็น code ได้ดังนี้</p> <pre>struct date {     int day; //วันเกิด     char month[10]; //เดือนเกิด     int year; //ปีเกิด }; struct student {     int id;     char name[30];     struct date birthdate; // เก็บโครงสร้างวันเดือนปีเกิด };</pre>			
<p><b>ตัวอย่างที่ 3</b> การใช้งานตัวแปรแบบโครงสร้างซ้อนโครงสร้าง</p> <pre>1 #include &lt;stdio.h&gt; 2 int main() 3 { 4     struct date 5     { 6         int day; //วันเกิด 7         char month[10]; //เดือนเกิด</pre>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	11
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

ตัวอย่างที่ 3 การใช้งานตัวแปรแบบโครงสร้างซ้อนโครงสร้าง (ต่อ)

```

8      int year; //ปีเกิด
9      };
10     struct student
11     {
12         int id;
13         char name[30];
14         struct date birthDate; // เก็บโครงสร้างวันเดือนปีเกิด
15     }std ;
16     printf("Input student id : ");
17     scanf("%d", &std.id);
18     printf("Input student name : ");
19     scanf("%s", &std.name);
20     printf("Input student birth date (day) : ");
21     scanf("%d", &std.birthDate.day);
22     printf("Input student birth date (month) : ");
23     scanf("%s", &std.birthDate.month);
24     printf("Input student birth date (year) : ");
25     scanf("%d", &std.birthDate.year);
26     printf("student id : %d\n", std.id);
27     printf("student name : %s\n", std.name);
28     printf("student birth date : %d/%s/%d", std.birthDate.day, std.birthDate.month,
std.birthDate.year);
29 }

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	11
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

ผลการรันโปรแกรม :

```

Input student id : 12345
Input student name : sunee
Input student birth date <day> : 22
Input student birth date <month> : 06
Input student birth date <year> : 2550
student id : 12345
student name : sunee
student birth date : 22/06/2550

```

### 5. ข้อมูลแบบยูเนียน

ในการกำหนดตัวแปรตามปกติ ตัวแปรแต่ละตัวจะมีพื้นที่เก็บข้อมูลแยกจากกัน แต่ยูเนียนเป็นการกำหนดโครงสร้างที่ทำให้ตัวแปรหรือกลุ่มตัวแปรที่กำหนดนั้นมาใช้พื้นที่เดียวกัน ยูเนียนคล้ายกับสตรักเจอร์ คือ ยูเนียนประกอบด้วยสมาชิกหลายตัวซึ่งอาจมีข้อมูลที่แตกต่างกัน แต่สมาชิกทุกตัวในยูเนียนจะใช้เนื้อที่หน่วยความจำของคอมพิวเตอร์ร่วมกัน ในขณะที่สมาชิกแต่ละตัวของสตรักเจอร์จะมีหน่วยความจำของตัวเอง ดังนั้น การใช้ยูเนียนสามารถประหยัดหน่วยความจำกับการทำงานที่ต้องใช้สมาชิกหลายตัว แต่ไม่จำเป็นต้องกำหนดค่าให้กับสมาชิกทุกตัวในเวลาเดียวกัน

ในการจัดเก็บข้อมูลของสมาชิกที่มีชนิดต่างกัน ใช้จำนวนเนื้อที่หน่วยความจำไม่เท่ากัน โปรแกรม จะกำหนดตัวแปรให้เองโดยอัตโนมัติ แต่อย่างไรก็ตามควรจะทราบว่าในขณะนั้นเก็บข้อมูลชนิดใดในยูเนียน เนื่องจากการเรียกใช้ข้อมูลผิดชนิด จะได้ผลลัพธ์ที่ไม่มีความหมาย รูปแบบทั่วไปในการกำหนดยูเนียน มี 2 รูปแบบ คือ ประกาศโครงสร้างข้อมูลก่อนแล้วกำหนดตัวแปรยูเนียนภายหลังกับประกาศโครงสร้าง ข้อมูลยูเนียนและกำหนดตัวแปรยูเนียนพร้อมกัน โดยสามารถกำหนดตัวแปรชนิดยูเนียนได้ตามรูปแบบ ต่อไปนี้

```

union union_name
{
    type1 name 1;
    type2 name 2;
    :
    type n name n;
} union_variable 1, ..., union_variable n;

```

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	11
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

เช่น

union member

{

char name[15];

char surname[15];

int year;

float salary;

} teacher, student;

**ตัวอย่างที่ 4** การเปรียบเทียบขนาดหน่วยความจำแบบสตรักเจอร์กับยูเนียน

```

1 #include <stdio.h>
2 union job { //defining a union
3     char name[32];
4     float salary;
5     int worker_no;
6 }u;
7 struct job1 {
8     char name[32];
9     float salary;
10    int worker_no;
11 }s;
12 int main(){
13     printf("Size of union = %d", sizeof(u));
14     printf("\nSize of structure = %d", sizeof(s));
15     return 0;
16 }
```

A = 200, B = 400

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>11</b>
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูชักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>11</b>
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>11</b>
	ชื่อหน่วย ข้อมูลชนิดโครงสร้างและยูเนียน	สอนครั้งที่	17
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>12</b>
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>1. สาระสำคัญ</b></p> <p>การประกอบวงจรอิเล็กทรอนิกส์จะต้องนำอุปกรณ์ต่างๆ เช่น ตัวต้านทาน ตัวเก็บประจุไดโอด ทรานซิสเตอร์ มาต่อกัน โดยขาของอุปกรณ์จะถูกบัดกรีให้เชื่อมต่อถึงกัน การบัดกรีเป็นทักษะที่สามารถพัฒนาได้โดยการฝึกปฏิบัติและการมีความรู้ที่ถูกต้อง</p> <p><b>2. สมรรถนะประจำหน่วย</b></p> <p>2.1. แสดงความรู้เกี่ยวกับภาษาซีกับไฟล์ข้อมูล</p> <p><b>3. จุดประสงค์การเรียนรู้</b></p> <p><b>3.1. ด้านความรู้</b></p> <p>3.1.1 แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรมด้วยคอมพิวเตอร์โดยใช้ภาษาซี หรือโปรแกรมภาษาอื่น ๆ</p> <p>3.1.2 เขียนโปรแกรมด้วยคอมพิวเตอร์</p> <p><b>3.2 ด้านทักษะ</b></p> <p>-</p> <p><b>3.3 คุณลักษณะที่พึงประสงค์</b></p> <p>3.3.1 มีเหตุผล</p> <p>3.3.2 ใฝ่เรียนรู้</p> <p>3.3.3 รู้จักกาลเทศะ</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	12
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

#### 4. เนื้อหาสาระการเรียนรู้

##### ภาษาซีกับไฟล์ข้อมูล

##### 1. ไฟล์ข้อมูล

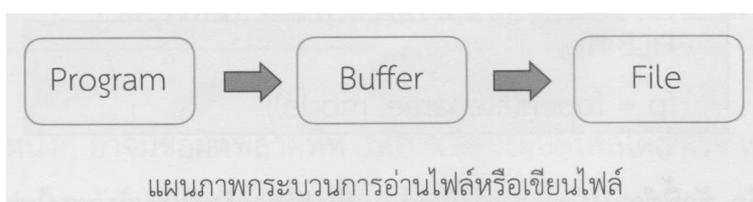
ไฟล์ (File) คือ ที่เก็บข้อมูลต่างๆ ของคอมพิวเตอร์ ในภาษาซีนั้นไฟล์จะมีความหมายรวมถึงอุปกรณ์ต่างๆ ที่ต่อกับเครื่องคอมพิวเตอร์ด้วย เช่น คีย์บอร์ด จอภาพ เป็นต้น การติดต่อกับไฟล์จะต้องผ่านลอจิคอลอินเตอร์เฟซ (Logical Interfaces) ที่เรียกว่าสตรีม (Stream) สตรีมจะช่วยให้ผู้เขียนโปรแกรมติดต่อกับอุปกรณ์ต่างๆ ได้ ซึ่งสตรีมที่ใช้ติดต่อกับไฟล์จะมีอยู่ 2 ประเภท คือ

1.1 Text File เป็นไฟล์ของตัวอักษร เพราะมีโครงสร้างในการเก็บข้อมูลเป็นตัวอักษร ไฟล์นั้น จึงไม่สามารถที่จะเก็บข้อมูลค่าตัวเลขจำนวนเต็ม จุดทศนิยม หรือในลักษณะที่เป็นโครงสร้าง ซึ่งการเก็บ ข้อมูลจะถูกแปลงเป็นเลขฐานสองตามรหัส ASCII

1.2 Binary File เป็นไฟล์ที่เก็บข้อมูลที่อยู่ในรูปแบบของค่าตรงๆ ซึ่งข้อมูลที่ไฟล์ประเภทนี้ จัดเก็บจะสามารถเป็นได้ทั้งตัวเลขจำนวนเต็ม ตัวเลขทศนิยม ตัวอักษร อาร์เรย์ และข้อมูลแบบโครงสร้าง โดยการจัดเก็บนั้นจะเก็บตามข้อมูลที่รับเข้ามาโดยไม่มีการแปลงค่าใดๆ

##### 2. การอ่านไฟล์และการเขียนไฟล์

หากสั่งให้โปรแกรมที่สร้างขึ้นมามีอ่านไฟล์ข้อมูลจะถูกอ่านขึ้นมาเก็บไว้ในหน่วยความจำก่อนนำไปใช้ หากสั่งให้โปรแกรมที่สร้างขึ้นมาเขียนไฟล์ ข้อมูลที่จะเขียนจะถูกเก็บไว้ในหน่วยความจำก่อนจะถูกเขียน ลงไฟล์ เรียกหน่วยความจำที่พักนี้ว่า บัฟเฟอร์ (Buffer)



##### 3. ขั้นตอนการทำงานร่วมกับไฟล์ข้อมูล

ในการกระทำกับไฟล์ได้แก่ การแก้ไข การเพิ่ม ไม่ได้กระทำกับไฟล์ที่เก็บอยู่ในดิสก์หรือฮาร์ดดิสก์โดยตรง แต่กระทำกับพื้นที่ในหน่วยความจำ (Memory) ซึ่งเครื่องคอมพิวเตอร์จัดเตรียมไว้เพื่อประมวลผลเกี่ยวกับไฟล์ หน่วยความจำที่จัดเตรียมไว้เพื่อเป็นตัวกลางในการประมวลผลกับไฟล์นี้เรียกว่า บัฟเฟอร์ เช่น ถ้าจะกระทำกับไฟล์ 3 ไฟล์ คือ ไฟล์ A, B, C คอมพิวเตอร์จะเตรียมบัฟเฟอร์ไว้ 3 ส่วน สำหรับแต่ละไฟล์ ไฟล์ในภาษาซีมีการจัดเก็บในลักษณะเรียงต่อกันไปตั้งแต่ต้นจนจบไม่มีการแบ่งเป็นช่วง ดังนั้นการจะกระทำ

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b> 12																							
	<b>ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล</b>	<b>สอนครั้งที่</b> 18																							
		<b>ชั่วโมงรวม</b> 72																							
		<b>จำนวนชั่วโมง</b> 4																							
<p>กับข้อมูลในไฟล์จะต้องระบุตำแหน่ง สิ่งที่ใช้ในการหาตำแหน่งของข้อมูลในไฟล์ข้อมูลเรียกว่าตัวชี้ไฟล์ (File Pointer) โดยเมื่อเปิดไฟล์จะต้องมีการสร้างตัวชี้ไฟล์หรือไฟล์พอยน์เตอร์ ซึ่งจะชี้ไปยังตำแหน่งเริ่มต้นของไฟล์ข้อมูลและเปลี่ยนตำแหน่งไปยังตำแหน่งที่ถูกระบุ ในการประมวลผลหลายไฟล์ข้อมูล แต่ละไฟล์ข้อมูล จะมีตัวชี้ตำแหน่งเป็นของตัวเองไม่ได้ใช้ร่วมกับไฟล์ข้อมูลอื่น เมื่อเริ่มทำงานกับภาษาซีระบบปฏิบัติการจะ เปิดไฟล์ให้โดยอัตโนมัติ 3 ไฟล์ คือ ไฟล์นำเข้ามาตรฐาน (stdin) ไฟล์ส่งออกมาตรฐาน (stdout) และไฟล์ ข้อผิดพลาดมาตรฐาน (stderr) สามารถเข้าถึงแต่ละระเบียนในไฟล์ได้ทั้งแบบเรียงลำดับและแบบโดยตรง โดยใช้ตัวชี้ไฟล์ (File Pointer) เป็นตัวระบุตำแหน่งในการทำงานกับไฟล์หนึ่งๆ จะต้องประกอบด้วยขั้นตอน ทั่วไป ดังนี้</p> <ol style="list-style-type: none"> <li>1) การเปิดไฟล์ข้อมูล เมื่อต้องการเริ่มต้นทำงานกับไฟล์</li> <li>2) การประมวลผลข้อมูลในไฟล์ข้อมูล คือ การเขียนข้อมูลหรืออ่านข้อมูลจากไฟล์</li> <li>3) การปิดไฟล์ข้อมูล เมื่อต้องการจะเลิกทำงานกับไฟล์</li> </ol>																									
<table border="1"> <thead> <tr> <th colspan="2">Mode</th> <th rowspan="2">ความหมาย</th> </tr> <tr> <th>Text</th> <th>Binary File</th> </tr> </thead> <tbody> <tr> <td>"r"</td> <td>"rb"</td> <td>เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านข้อมูลขึ้นมาจากแฟ้มอย่างเดียว ถ้าไฟล์ข้อมูลที่ระบุให้เป็นไฟล์ใหม่จะไม่สามารถเปิดไฟล์ได้</td> </tr> <tr> <td>"w"</td> <td>"wb"</td> <td>เปิดไฟล์ข้อมูลใหม่เพื่อบันทึกข้อมูลลงแฟ้มอย่างเดียว ถ้าไฟล์ข้อมูลที่ระบุ ให้เป็นไฟล์เก่าข้อมูลในไฟล์เก่าจะถูกลบทิ้งไปโดยอัตโนมัติพร้อมกับสร้าง ไฟล์ใหม่ในชื่อเดิม</td> </tr> <tr> <td>"a"</td> <td>"ab"</td> <td>เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้เพื่อบันทึกข้อมูลเพิ่มเติมลงไปไฟล์ข้อมูลเก่า โดยข้อมูลที่เพิ่มเข้าไปจะไปอยู่ตอนท้ายของไฟล์เสมอ ถ้าไฟล์ข้อมูลที่ระบุ ให้เป็นไฟล์ใหม่จะเป็นการสร้างไฟล์ใหม่ แต่บันทึกข้อมูลได้ครั้งเดียวเท่านั้น</td> </tr> <tr> <td>"r + 1"</td> <td>"r + b"</td> <td>เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านและบันทึกข้อมูลในแฟ้ม ถ้าใช้โหมดนี้เพื่อบันทึกข้อมูลลงไฟล์และใช้คู่กับคำสั่งบันทึกข้อมูลลงไฟล์ จะทำให้บันทึกข้อมูลใหม่ทับข้อมูลเก่าลงไปตั้งแต่ต้นไฟล์ไปจนถึงท้ายไฟล์ ทำให้ข้อมูล เดิมถูกแทนที่ด้วยข้อมูลใหม่</td> </tr> <tr> <td>"w + 1"</td> <td>"w + b"</td> <td>เปิดไฟล์ข้อมูลใหม่เพื่อบันทึกและอ่านข้อมูลในไฟล์ ถ้าแฟ้มข้อมูลที่ระบุไว้ เป็นไฟล์ที่กำลังใช้งานอยู่ จะเป็นการทำลายข้อมูลในไฟล์ทั้งหมดและสร้าง ไฟล์ใหม่ขึ้นมาแทนที่</td> </tr> <tr> <td>"a + 1"</td> <td>"a + b"</td> <td>เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านข้อมูลจากไฟล์และเพิ่มข้อมูล เข้าไปในไฟล์ข้อมูล โดยข้อมูลที่เพิ่มเข้าไปจะอยู่ตอนท้ายไฟล์ข้อมูลเสมอ ถ้า ไฟล์ข้อมูลที่ระบุให้เป็นไฟล์ใหม่ จะเป็นการสร้างไฟล์ข้อมูลใหม่ตามชื่อที่ ระบุไว้ทันที และสามารถบันทึกข้อมูลลงไปในไฟล์ได้</td> </tr> </tbody> </table>			Mode		ความหมาย	Text	Binary File	"r"	"rb"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านข้อมูลขึ้นมาจากแฟ้มอย่างเดียว ถ้าไฟล์ข้อมูลที่ระบุให้เป็นไฟล์ใหม่จะไม่สามารถเปิดไฟล์ได้	"w"	"wb"	เปิดไฟล์ข้อมูลใหม่เพื่อบันทึกข้อมูลลงแฟ้มอย่างเดียว ถ้าไฟล์ข้อมูลที่ระบุ ให้เป็นไฟล์เก่าข้อมูลในไฟล์เก่าจะถูกลบทิ้งไปโดยอัตโนมัติพร้อมกับสร้าง ไฟล์ใหม่ในชื่อเดิม	"a"	"ab"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้เพื่อบันทึกข้อมูลเพิ่มเติมลงไปไฟล์ข้อมูลเก่า โดยข้อมูลที่เพิ่มเข้าไปจะไปอยู่ตอนท้ายของไฟล์เสมอ ถ้าไฟล์ข้อมูลที่ระบุ ให้เป็นไฟล์ใหม่จะเป็นการสร้างไฟล์ใหม่ แต่บันทึกข้อมูลได้ครั้งเดียวเท่านั้น	"r + 1"	"r + b"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านและบันทึกข้อมูลในแฟ้ม ถ้าใช้โหมดนี้เพื่อบันทึกข้อมูลลงไฟล์และใช้คู่กับคำสั่งบันทึกข้อมูลลงไฟล์ จะทำให้บันทึกข้อมูลใหม่ทับข้อมูลเก่าลงไปตั้งแต่ต้นไฟล์ไปจนถึงท้ายไฟล์ ทำให้ข้อมูล เดิมถูกแทนที่ด้วยข้อมูลใหม่	"w + 1"	"w + b"	เปิดไฟล์ข้อมูลใหม่เพื่อบันทึกและอ่านข้อมูลในไฟล์ ถ้าแฟ้มข้อมูลที่ระบุไว้ เป็นไฟล์ที่กำลังใช้งานอยู่ จะเป็นการทำลายข้อมูลในไฟล์ทั้งหมดและสร้าง ไฟล์ใหม่ขึ้นมาแทนที่	"a + 1"	"a + b"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านข้อมูลจากไฟล์และเพิ่มข้อมูล เข้าไปในไฟล์ข้อมูล โดยข้อมูลที่เพิ่มเข้าไปจะอยู่ตอนท้ายไฟล์ข้อมูลเสมอ ถ้า ไฟล์ข้อมูลที่ระบุให้เป็นไฟล์ใหม่ จะเป็นการสร้างไฟล์ข้อมูลใหม่ตามชื่อที่ ระบุไว้ทันที และสามารถบันทึกข้อมูลลงไปในไฟล์ได้
Mode		ความหมาย																							
Text	Binary File																								
"r"	"rb"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านข้อมูลขึ้นมาจากแฟ้มอย่างเดียว ถ้าไฟล์ข้อมูลที่ระบุให้เป็นไฟล์ใหม่จะไม่สามารถเปิดไฟล์ได้																							
"w"	"wb"	เปิดไฟล์ข้อมูลใหม่เพื่อบันทึกข้อมูลลงแฟ้มอย่างเดียว ถ้าไฟล์ข้อมูลที่ระบุ ให้เป็นไฟล์เก่าข้อมูลในไฟล์เก่าจะถูกลบทิ้งไปโดยอัตโนมัติพร้อมกับสร้าง ไฟล์ใหม่ในชื่อเดิม																							
"a"	"ab"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้เพื่อบันทึกข้อมูลเพิ่มเติมลงไปไฟล์ข้อมูลเก่า โดยข้อมูลที่เพิ่มเข้าไปจะไปอยู่ตอนท้ายของไฟล์เสมอ ถ้าไฟล์ข้อมูลที่ระบุ ให้เป็นไฟล์ใหม่จะเป็นการสร้างไฟล์ใหม่ แต่บันทึกข้อมูลได้ครั้งเดียวเท่านั้น																							
"r + 1"	"r + b"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านและบันทึกข้อมูลในแฟ้ม ถ้าใช้โหมดนี้เพื่อบันทึกข้อมูลลงไฟล์และใช้คู่กับคำสั่งบันทึกข้อมูลลงไฟล์ จะทำให้บันทึกข้อมูลใหม่ทับข้อมูลเก่าลงไปตั้งแต่ต้นไฟล์ไปจนถึงท้ายไฟล์ ทำให้ข้อมูล เดิมถูกแทนที่ด้วยข้อมูลใหม่																							
"w + 1"	"w + b"	เปิดไฟล์ข้อมูลใหม่เพื่อบันทึกและอ่านข้อมูลในไฟล์ ถ้าแฟ้มข้อมูลที่ระบุไว้ เป็นไฟล์ที่กำลังใช้งานอยู่ จะเป็นการทำลายข้อมูลในไฟล์ทั้งหมดและสร้าง ไฟล์ใหม่ขึ้นมาแทนที่																							
"a + 1"	"a + b"	เปิดไฟล์ข้อมูลเก่าที่เคยสร้างไว้แล้วเพื่ออ่านข้อมูลจากไฟล์และเพิ่มข้อมูล เข้าไปในไฟล์ข้อมูล โดยข้อมูลที่เพิ่มเข้าไปจะอยู่ตอนท้ายไฟล์ข้อมูลเสมอ ถ้า ไฟล์ข้อมูลที่ระบุให้เป็นไฟล์ใหม่ จะเป็นการสร้างไฟล์ข้อมูลใหม่ตามชื่อที่ ระบุไว้ทันที และสามารถบันทึกข้อมูลลงไปในไฟล์ได้																							

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>12</b>
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	<b>สอนครั้งที่</b>	18
		<b>ชั่วโมงรวม</b>	72
		<b>จำนวนชั่วโมง</b>	4
<p>การเปิดไฟล์หนึ่งๆ อาจมีข้อผิดพลาดที่ทำให้ไม่สามารถเปิดไฟล์ได้สำเร็จ เช่น</p> <ol style="list-style-type: none"> <li>1) มีการป้องกันการบันทึกข้อมูลทับ (Write Protect)</li> <li>2) เนื้อที่ในหน่วยความจำสำรองเต็ม</li> <li>3) ไม่มีชื่อไฟล์ที่ระบุ (ในกรณีที่เปิดไฟล์เพื่ออ่าน)</li> </ol> <p>ในกรณีเหล่านี้คำสั่ง fopen() จะคืนค่า NULL ถ้าไม่สามารถเปิดไฟล์ได้ แต่ถ้าเปิดไฟล์สำเร็จจะ คืนค่าตำแหน่ง</p> <p><b>ตัวอย่างที่ 1</b> ส่วนของโปรแกรมแสดงการเปิดไฟล์เพื่อเขียนข้อมูลลงไฟล์</p> <pre>FILE *fp; fp = fopen("H:\\lab_c.txt", "w");</pre> <p><b>คำอธิบาย</b> เปิดไฟล์ข้อมูลชื่อ lab_c.txt ที่อยู่ไดรฟ์ H เพื่อใช้สำหรับบันทึกข้อมูลลงไฟล์อย่างเดียว โดยมี ตัวแปร fp เป็น File Pointer ซึ่งไปยังตำแหน่งของข้อมูลในไฟล์</p> <p><b>ตัวอย่างที่ 2</b> ส่วนของโปรแกรมแสดงการเปิดไฟล์เพื่ออ่านไฟล์</p> <pre>FILE *fp; fp = fopen("C:\\lab_c.txt", "r"); if (fp == NULL) printf("file doesn't exist \n");</pre> <p><b>คำอธิบาย</b> เปิดไฟล์ข้อมูลชื่อ lab_c.txt ที่อยู่ไดรฟ์ H เพื่อใช้สำหรับอ่านข้อมูลจากไฟล์ โดยมีตัวแปร fp เป็น File Pointer ซึ่งไปยังตำแหน่งของข้อมูลในไฟล์ ถ้าไม่มีไฟล์ชื่อ lab_c.txt โปรแกรมจะแสดงข้อความ "File doesn't exist." ให้ผู้ใช้ทราบ</p> <p><b>5. คำสั่งปิดไฟล์ข้อมูล</b></p> <p>หลังจากที่สิ้นสุดการใช้งานไฟล์หนึ่งๆ แล้ว ผู้ใช้ควรที่จะทำการปิดไฟล์เพื่อให้ระบบนำข้อมูลที่เหลือในบัฟเฟอร์ทั้งหมดเข้าไปบันทึกในไฟล์และป้องกันข้อมูลไม่ให้สูญหาย รูปแบบคำสั่งที่ใช้ในการปิดไฟล์ มีดังนี้</p> <p style="text-align: center;"><b>fclose (file pointer);</b></p> <p><b>คำอธิบาย</b></p> <p>file pointer คือ ชื่อของตัวชี้ไฟล์ (File Pointer) ที่ต้องการจะปิด ถ้าการปิดไฟล์ด้วยคำสั่ง fclose() สำเร็จจะคืนค่าศูนย์ (0) แต่ถ้าการปิดไฟล์ไม่สำเร็จจะคืนค่าอื่นๆ ตัวอย่างที่ 3 ส่วนของโปรแกรมแสดงการเปิดไฟล์เพื่อปิดไฟล์ MyFile.txt</p> <pre>FILE *fp;</pre>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>12</b>
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<pre>fp = fopen("C:\\MyFile.txt", "r"); fclose(fp);</pre> <p><b>6. คำสั่งอ่านและเขียนอักขระ</b></p> <p>การอ่านและเขียนอักขระเป็นการประมวลผลทีละตัวอักขระ โดยใช้คำสั่ง <code>putcO</code> และ <code>getcO</code> ทั้งนี้คำสั่งทั้งสองจะมีพารามิเตอร์ตัวชี้ไฟล์เพื่อที่จะได้รู้ว่าจะทำงานกับไฟล์ใด</p> <p>6.1 คำสั่ง <code>getc()</code> คือ คำสั่งอ่านอักขระตัวต่อไปในสายอักขระภายในไฟล์ ซึ่งถูกอ้างอิงด้วย ตัวชี้ไฟล์จนพบ <code>&lt;EOF&gt;</code> เมื่อจบไฟล์ รูปแบบคำสั่งที่ใช้มีดังนี้</p> <p style="text-align: center;"><code>getc(fp);</code> หรือ <code>single_char = getc(fp);</code></p> <p><b>คำอธิบาย</b></p> <p><code>single_char</code> คือ ตัวแปรชนิด Single Character ที่ใช้เก็บข้อมูลตัวอักขระตัวเดียว <code>fp</code> คือ ตัวชี้ของไฟล์ข้อมูล</p> <p>6.2 คำสั่ง <code>putc()</code> คือ คำสั่งการนำอักขระในสายอักขระที่กำหนด เขียนลงในไฟล์ซึ่งถูกอ้างอิง ด้วยตัวชี้ไฟล์</p> <p style="text-align: center;"><code>putc(single_char, fp);</code></p> <p><b>คำอธิบาย</b></p> <p><code>single_char</code> คือ ตัวแปรชนิด Single Character ที่ใช้เก็บข้อมูลตัวอักขระตัวเดียว <code>fp</code> คือ ตัวชี้ของไฟล์ข้อมูล</p> <p><b>ตัวอย่างที่ 4</b> โปรแกรมอ่านตัวอักขระจากไฟล์แล้วแสดงผลอักขระออกทางหน้าจอ</p> <pre>1 #include &lt;stdio.h&gt; 2 int main() { 3     FILE *fp; 4     char c; 5     fp = fopen("C:\\LAB_C.txt", "r"); 6     if (fp == NULL) { 7         printf("File doesn't exist\n"); 8     }</pre>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	12
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

ตัวอย่างที่ 4 โปรแกรมอ่านตัวอักขระจากไฟล์แล้วแสดงตัวผลอักขระออกทางหน้าจอ (ต่อ)

```

9   else {
10      do{
11         c = getc(fp); //อ่านอักขระจากไฟล์
12         putchar(c); //แสดงอักขระบนจอภาพ
13      }while(c != EOF); //วนซ้ำจนจบไฟล์
14   }
15   fclose(fp);
16   return 0;
17 }
```

#### 7. คำสั่งที่ใช้ในการอ่านและเขียนข้อมูลไฟล์

7.1 คำสั่ง fscanf() คือ คำสั่งที่ใช้ในการอ่านข้อมูลจากไฟล์ข้อมูลแล้วนำมาเก็บไว้ในตัวแปรที่ต้องการโดยมีการทำงานคล้ายคำสั่ง scanf() แตกต่างกันตรงที่คำสั่ง fscanf() เป็นคำสั่งที่ใช้อ่านข้อมูลจากไฟล์ข้อมูล แต่คำสั่ง scanf() ใช้รับข้อมูลจากคีย์บอร์ดแล้วไปเก็บไว้ในตัวแปรที่ต้องการ รูปแบบคำสั่งที่ใช้มี ดังนี้

**fscanf(fp, control string, variable list);**

fp คือ File Pointer ของไฟล์ข้อมูล

control string คือ รหัสรูปแบบของข้อมูลและรหัสควบคุมใช้เหมือนคำสั่ง scanf() เช่น %d %s

เป็นต้น

variable list คือ ชื่อตัวแปรที่ใช้เก็บข้อมูลที่อ่านมาจากไฟล์ข้อมูล โดยจะต้องระบุเครื่องหมาย & (Ampersand) นำหน้าชื่อตัวแปรด้วย ยกเว้นตัวแปร String เท่านั้นที่ไม่ต้องมีเครื่องหมาย &

7.2 คำสั่ง fprintf() คือ คำสั่งที่ใช้บันทึกข้อมูลลงไฟล์ โดยสามารถจัดรูปแบบข้อมูลที่ต้องการ บันทึกได้คล้ายกับคำสั่ง printf() แตกต่างกันตรงที่ printf() เป็นคำสั่งที่ใช้พิมพ์ผลลัพธ์ออกทางจอภาพ แต่ คำสั่ง fprintf() ใช้บันทึกข้อมูลลงไฟล์ รูปแบบคำสั่งที่ใช้มีดังนี้

**fprintf(fp, control string, variable list);**

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	12
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4

fp คือ file pointer ของไฟล์ข้อมูล

control string คือ รหัสรูปแบบของข้อมูลและรหัสควบคุมใช้เหมือนคำสั่ง printf() เช่น %d %r เป็นต้น

variable list คือ ค่าคงที่ ตัวแปร หรือนิพจน์ที่จะเขียนลงไฟล์ข้อมูล แต่ถ้าเป็นค่า string ต้อง เขียนอยู่ในเครื่องหมาย ""

**ตัวอย่างที่ 5** โปรแกรมแสดงการเขียนข้อมูลลงไฟล์ File.txt

```

1  #include <stdio.h>
2  int main()
3  {
4      FILE *stream;
5      int i = 100;
6      char c = 'C';
7      float f = 1.234;
8      /* open a file for update */
9      stream = fopen("C:\\File.txt", "w+");
10     /* write some data to the file */
11     fprintf(stream, "%d %c %f", i, c, f);
12     /* close the file */
13     fclose(stream);
14     return 0;
15 }
```

7.3 คำสั่ง rewind() เป็นคำสั่งที่ใช้ย้ายตำแหน่งของ File Pointer ไปยังตำแหน่งเริ่มต้นของไฟล์ รูปแบบคำสั่งมีดังนี้

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>12</b>												
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	<b>สอนครั้งที่</b>	<b>18</b>												
		<b>ชั่วโมงรวม</b>	<b>72</b>												
		<b>จำนวนชั่วโมง</b>	<b>4</b>												
<pre>rewind(fp);</pre> <p>7.4 คำสั่ง fseek() เป็นคำสั่งที่ข้าย้ายตำแหน่งของ File Pointer ไปยังตำแหน่งที่ต้องการในไฟล์ข้อมูล โดยจะต้องกำหนดจุดเริ่มต้น (Origin) ของ File Pointer และค่า Offset รูปแบบคำสั่งมีดังนี้</p> <p style="text-align: center;"><b>fseek(fp, offset, origing);</b></p> <p>offset คือ ระยะห่างจากตำแหน่งจุดเริ่มต้น มีหน่วยเป็น byte origin คือ จุดที่ File Pointer ชี้อยู่ มีอยู่ 3 สถานะ ดังตารางที่ 12.2</p> <p><b>ตารางที่ 12.2</b> สถานะจุดที่ File Point ชี</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #d3d3d3;">Macro Name</th> <th style="background-color: #d3d3d3;">ค่าคงที่</th> <th style="background-color: #d3d3d3;">ความหมาย</th> </tr> </thead> <tbody> <tr> <td>SEEK_SET</td> <td style="text-align: center;">0</td> <td>File Pointer อยู่ที่ต้นไฟล์ (BOF)</td> </tr> <tr> <td>SEEK_CUR</td> <td style="text-align: center;">1</td> <td>File Pointer อยู่ที่ตำแหน่งปัจจุบัน</td> </tr> <tr> <td>SEEK_END</td> <td style="text-align: center;">2</td> <td>File Pointer อยู่ที่ท้ายไฟล์ (EOF)</td> </tr> </tbody> </table> <p>7.5 คำสั่ง ftell() เป็นคำสั่งที่ชี้บอกตำแหน่งของ File Pointer ว่าปัจจุบันชี้อยู่ที่ตำแหน่งใดใน ไฟล์ข้อมูล โดยคำสั่งนี้จะส่งค่ากลับเป็นตัวเลขจำนวนเต็ม รูปแบบคำสั่งมีดังนี้</p> <p style="text-align: center;"><b>ftell(fp, offset, origin);</b></p> <p>7.6 คำสั่ง remove() คำสั่งใช้ลบไฟล์ รูปแบบ คือ remove(file name) เช่น</p> <pre>result = remove("c:\\text4.txt"); /* result เป็นชื่อตัวแปร */</pre> <p><b>ข้อควรจำ</b> ฟังก์ชันนี้เมื่อลบไฟล์สำเร็จจะให้ค่าเป็น 0 หรือเท็จกลับมา ถ้าลบไม่สำเร็จจะให้ค่าอื่น ที่ไม่ใช่ 0 ดังนั้น การเขียนโปรแกรมเพื่อตรวจสอบจึงต้องระวัง</p> <p>7.7 คำสั่ง rename() คำสั่งใช้เปลี่ยนชื่อไฟล์ รูปแบบ คือ rename(old name, new name)</p> <pre>result = rename("c:\\c_train\\t4.txt", "c:\\c_train\\b4.dat");</pre>				Macro Name	ค่าคงที่	ความหมาย	SEEK_SET	0	File Pointer อยู่ที่ต้นไฟล์ (BOF)	SEEK_CUR	1	File Pointer อยู่ที่ตำแหน่งปัจจุบัน	SEEK_END	2	File Pointer อยู่ที่ท้ายไฟล์ (EOF)
Macro Name	ค่าคงที่	ความหมาย													
SEEK_SET	0	File Pointer อยู่ที่ต้นไฟล์ (BOF)													
SEEK_CUR	1	File Pointer อยู่ที่ตำแหน่งปัจจุบัน													
SEEK_END	2	File Pointer อยู่ที่ท้ายไฟล์ (EOF)													

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>12</b>
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	<b>สอนครั้งที่</b>	<b>18</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>4</b>
<p><b>ข้อควรจำ</b> ฟังก์ชันนี้เมื่อเปลี่ยนชื่อไฟล์สำเร็จจะให้ค่าเป็น 0 หรือเท็จกลับมา ถ้าเปลี่ยนชื่อไม่สำเร็จ จะให้ค่าอื่นที่ไม่ใช่ 0 ดังนั้น การเขียนโปรแกรมเพื่อตรวจสอบจึงต้องระวัง</p> <p>7.8 คำสั่ง <code>ferror()</code> เป็นคำสั่งใช้ตรวจสอบว่ามีข้อผิดพลาดจากการอ่านไฟล์ข้อมูลหรือไม่ รูปแบบมีดังนี้</p> <p style="text-align: center;"><b><code>ferror(file pointer)</code></b></p> <p><b>ข้อควรจำ</b> ฟังก์ชันนี้ถ้าตรวจสอบว่าไม่มีข้อผิดพลาดจากการอ่านไฟล์จะให้ค่าเป็น 0 หรือเท็จ กลับมาถ้าพบข้อผิดพลาดจะให้ค่าอื่นที่ไม่ใช่ 0 ดังนั้น การเขียนโปรแกรมเพื่อตรวจสอบจึงต้องระวัง</p> <p>7.9 คำสั่ง <code>feof()</code> เป็นคำสั่งใช้ตรวจสอบว่าข้อมูลหมดไฟล์หรือไม่ เพื่อหาจุดสิ้นสุดของไฟล์ รูปแบบมีดังนี้</p> <p style="text-align: center;"><b><code>feof(file pointer)</code></b></p> <p>โดยเมื่อถึงจุดสิ้นสุดของไฟล์จะคืนค่าที่ไม่ใช่ 0 มาให้ และคืนค่า 0 เมื่อยังไม่ถึงจุดสิ้นสุดของไฟล์ การตรวจสอบโดยใช้ฟังก์ชันนี้ อาจดีกว่าการใช้ค่าคงที่ EOF สำหรับไฟล์ <code>stdin</code> หรือการรับข้อมูลจาก แป้นพิมพ์ รหัสแสดงว่าจบไฟล์ คือ การกดแป้น Ctrl ค้างไว้แล้วเคาะแป้น Z</p>			

	<b>แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ</b>	<b>หน่วยที่</b>	<b>12</b>
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	<b>สอนครั้งที่</b>	<b>18</b>
		<b>ชั่วโมงรวม</b>	<b>72</b>
		<b>จำนวนชั่วโมง</b>	<b>4</b>
<p><b>5.1 การนำเข้าสู่บทเรียน</b></p> <p>5.1.1 ครูทักทายและขานชื่อผู้เรียน</p> <p>5.1.2 ครูซักชวนพูดคุยความปลอดภัยในชีวิตประจำวัน</p> <p>5.1.3 ครูแจ้งหัวข้อ/ จุดประสงค์/ การประเมินผล</p> <p><b>5.2 การเรียนรู้</b></p> <p>5.2.1 ครูให้ทำแบบทดสอบก่อนเรียน</p> <p>5.2.2 ครูบรรยาย, อธิบาย ผู้เรียนซักถามแสดง ความคิดเห็น</p> <p><b>5.3 การสรุป</b></p> <p>5.3.1 ครูให้ทำแบบฝึกหัดในชั้นเรียน ซักถามและอภิปรายร่วมกัน</p> <p>5.3.2 มอบหมายงานให้ทำในห้องเรียน</p> <p>5.3.3 ทำแบบทดสอบหลังเรียน</p> <p><b>5.4 การวัดผลและประเมินผล</b></p> <p>5.4.1 ใบเช็คชื่อ</p> <p>5.4.2 แบบทดสอบ</p> <p>5.4.3 แบบสังเกตพฤติกรรม</p> <p>5.4.4 แบบฝึกหัด</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	12
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>6. สื่อการเรียนรู้/แหล่งการเรียนรู้</b></p> <p><b>6.1 สื่อสิ่งพิมพ์</b></p> <p>6.1.1 ใบความรู้</p> <p>6.1.2 หนังสือ</p> <p><b>6.2 สื่อโสตทัศน (ถ้ามี)</b></p> <p>6.2.1 มัลติมีเดีย</p> <p>6.2.2 Power Point</p> <p><b>6.3 หุ่นจำลองหรือของจริง (ถ้ามี)</b></p> <p>-</p> <p><b>6.4 อื่น ๆ (ถ้ามี)</b></p> <p>-</p> <p><b>7. เอกสารประกอบการจัดการเรียนรู้ (ใบความรู้ ใบงาน ใบมอบหมายงาน ฯลฯ)</b></p> <p>7.1.1 ใบความรู้</p> <p>7.1.2 ใบงาน แบบฝึกหัด</p> <p>7.1.3 หนังสือเรียน (Textbook)</p> <p><b>8. การบูรณาการ/ความสัมพันธ์กับวิชาอื่น</b></p> <p>8.1.1 บูรณาการงานสวนพฤกษศาสตร์โรงเรียน</p> <p>8.1.2 บูรณาการหลักปรัชญาเศรษฐกิจพอเพียง</p>			

	แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ	หน่วยที่	12
	ชื่อหน่วย ภาษาซีกับไฟล์ข้อมูล	สอนครั้งที่	18
		ชั่วโมงรวม	72
		จำนวนชั่วโมง	4
<p><b>9. การวัดและประเมินผล</b></p> <p><b>9.1 ก่อนเรียน</b></p> <p>9.1.1 ขานชื่อผู้เรียน, ตรวจสอบการตรงต่อเวลา, การแต่งกาย</p> <p><b>9.2 ขณะเรียน</b></p> <p>9.2.2 สังเกตพฤติกรรม, ความสนใจ, ความรับผิดชอบ</p> <p>9.2.3 ตรวจสอบแบบฝึกหัดในห้องเรียน</p> <p><b>9.3 หลังเรียน</b></p> <p>9.3.1 แบบทดสอบหลังเรียน ใช้เกณฑ์ 60%</p> <p><b>10. บันทึกหลังสอน</b></p> <p><b>10.1 ผลการใช้แผนการจัดการเรียนรู้</b></p> <p>10.1.1 เนื้อหาสาระในแผนการจัดการเรียนรู้ครอบคลุมตามจุดประสงค์ของแผนก</p> <p>10.1.2 กิจกรรมที่นำมาใช้ประกอบบทเรียนในแผนมีความสอดคล้องกับบทเรียน เนื้อหา และเหมาะสมกับผู้เรียน</p>			