

เอกสารประกอบการสอน
รายวิชาการโปรแกรมคอมพิวเตอร์



กิตติรัตน์ รุ่งรัตนอุบล

คณะเทคโนโลยีอุตสาหกรรมและอัญมณีศาสตร์

มหาวิทยาลัยราชภัฏรำไพพรรณี

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี
2557

เอกสารประกอบการสอน
รายวิชาการโปรแกรมคอมพิวเตอร์



กิตติรัตน์ รุ่งรัตนอุบล
วศ.ม. (วิศวกรรมเกษตร)

คณะเทคโนโลยีอุตสาหกรรมและอัญมณีศาสตร์

มหาวิทยาลัยราชภัฏรำไพพรรณี

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี
2557

คำนำ

เอกสารประกอบการสอนรหัสวิชา 6091102 เล่มนี้เขียนเพื่อใช้ประกอบการเรียนการสอนในรายวิชาการโปรแกรมคอมพิวเตอร์ ซึ่งผู้สอนนั้นสำเร็จการศึกษาในระดับปริญญาตรีอิเล็กทรอนิกส์กำลังและปริญญาโทวิศวกรรมเกษตร จึงมีความสามารถในการนำระบบอิเล็กทรอนิกส์เข้ามาใช้ในการควบคุมเครื่องจักรกลทางการเกษตรและระบบอัตโนมัติต่าง ๆ หรือการใช้โปรแกรมคอมพิวเตอร์ในการควบคุม ทำให้ตัวผู้สอนมีความรู้ในการนำไปใช้งานจริง ตลอดจนมีประสบการณ์ในการสอนในรายวิชานี้มาเป็นเวลาหลายปี อีกทั้งได้สะสมประสบการณ์ ความรู้ ความสามารถ เพื่อจะถ่ายทอดสู่ผู้ที่สนใจ โดยเอกสารนี้ประกอบด้วยเนื้อหาเรื่อง คอมพิวเตอร์เบื้องต้น การเขียนผังงาน การออกแบบหน้าจอ การใช้งานคอนโทรลเบื้องต้น โครงสร้างทางภาษา การเขียนแอปพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์ การออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรม

ผู้สอนขอขอบคุณเจ้าของเอกสารตำราทุกท่าน ที่ผู้สอนใช้ศึกษา ค้นคว้าและอ้างอิง ในการเขียนครั้งนี้และขอขอบคุณมหาวิทยาลัยราชภัฏรำไพพรรณีที่สนับสนุน ส่งเสริมให้ผู้สอนทำผลงานวิชาการ ซึ่งผู้สอนคาดหวังว่าเอกสารประกอบการสอนเล่มนี้ จะสามารถถูกนำไปใช้ให้เกิดประโยชน์ต่อผู้เรียน และผู้สนใจที่จะศึกษาไม่มากนักน้อย

กิตติรัตน์ รุ่งรัตนอุบล
พฤษภาคม 2557

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

สารบัญ

	หน้า
คำนำ	(1)
สารบัญ	(2)
สารบัญภาพ	(5)
สารบัญตาราง	(10)
แผนบริหารการสอนประจำรายวิชา	(11)
แผนบริหารการสอนประจำบทที่ 1	1
บทที่ 1 คอมพิวเตอร์เบื้องต้น	3
หลักการของคอมพิวเตอร์	3
ลักษณะข้อมูลที่ใช้ในคอมพิวเตอร์และข้อมูลชนิดโครงสร้างไฟล์	4
อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์	5
โปรแกรมวิซวลเบสิก	25
การติดตั้งโปรแกรมวิซวลเบสิกจากไมโครซอฟท์วิซวลสตูดิโอ	27
สรุป	31
แบบฝึกหัดบทที่ 1	32
เอกสารอ้างอิง	33
แผนบริหารการสอนประจำบทที่ 2	37
บทที่ 2 การเขียนผังงาน	39
หลักการเขียนผังงานเบื้องต้น	39
การเขียนผังงานแบบตามลำดับ	40
การเขียนผังงานแบบเลือกเส้นทาง	42
การเขียนผังงานแบบทำซ้ำ	51
สรุป	55
แบบฝึกหัดบทที่ 2	56
เอกสารอ้างอิง	57
แผนบริหารการสอนประจำบทที่ 3	59
บทที่ 3 การออกแบบหน้าจอ	61
การใช้แสง	61
การใช้สี ขาว-ดำ ในการออกแบบ	65
การมีพื้นที่ว่าง	68
การวางตัวหนังสือบนรูปภาพ	68
การเพิ่มหรือลด ความเด่นของตัวหนังสือ	70
การเลือกใช้ชุดแบบอักษรให้เหมาะสม	72
สรุป	73

สารบัญ (ต่อ)

	หน้า
แบบฝึกหัดบทที่ 3	74
เอกสารอ้างอิง	75
แผนบริหารการสอนประจำบทที่ 4	77
บทที่ 4 การใช้งานคอนโทรลเบื้องต้น	79
การสร้างโครงงานใหม่	79
คอนโทรลбутทอน	81
คอนโทรลโอวีลเซพ	83
คอนโทรลพิกเจอร์บ็อกซ์	86
คอนโทรลลาเบล	88
คอนโทรลเท็กซ์บ็อกซ์	91
คอนโทรลเช็คบ็อกซ์	94
คอนโทรลเรดิโอบัตทอน	94
คอนโทรลกรุปบ็อกซ์	95
คอนโทรลสโครลบาร์	95
คอนโทรลไทมเมอร์	99
การตั้งชื่อคอนโทรลและอ็อบเจกต์ตามคำแนะนำของไมโครซอฟท์	101
สรุป	103
แบบฝึกหัดบทที่ 4	104
เอกสารอ้างอิง	106
แผนบริหารการสอนประจำบทที่ 5	107
บทที่ 5 โครงสร้างทางภาษา	109
ตัวแปร	109
คำสั่งวน	109
ค่าคงที่	110
ชนิดของข้อมูล	111
การกำหนดค่าให้กับตัวแปร	113
โอเปอเรเตอร์	115
ลำดับในการทำงานของโอเปอเรเตอร์	117
การเขียนคำอธิบายโปรแกรม	118
คำสั่งในการเขียนโปรแกรม	119
การรับข้อมูลและแสดงข้อความ	119
สรุป	120
แบบฝึกหัดบทที่ 5	121
เอกสารอ้างอิง	122

สารบัญ (ต่อ)

	หน้า
แผนบริหารการสอนประจำบทที่ 6	123
บทที่ 6 การเขียนแอปพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์	125
คำสั่งควบคุมการทำงานแบบลำดับ	125
คำสั่งควบคุมการทำงานแบบเลือกเส้นทาง	128
คำสั่งควบคุมการทำงานแบบทำซ้ำ	141
สรุป	161
แบบฝึกหัดบทที่ 6	162
เอกสารอ้างอิง	163
แผนบริหารการสอนประจำบทที่ 7	165
บทที่ 7 การออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรม	167
โปรแกรมเครื่องคิดเลข	167
โปรแกรมตั้งเวลาปลุก	171
โปรแกรมแสดงตัวอักษรวิ่ง	173
โปรแกรมบอร์ดควบคุมระดับน้ำ	174
โปรแกรมบอร์ดควบคุมไฟในถังผสม	178
โปรแกรมควบคุมไฟจราจร	184
สรุป	192
แบบฝึกหัดบทที่ 7	193
เอกสารอ้างอิง	194

สารบัญภาพ

ภาพที่	หน้า
1.1 กลุ่มของแป้นพิมพ์บนคีย์บอร์ดมาตรฐาน	6
1.2 คีย์บอร์ดติดตั้งภายในของโน้ตบุ๊ก	7
1.3 คีย์บอร์ดเออร์โกโนมิกส์	7
1.4 คีย์บอร์ดไร้สาย	7
1.5 คีย์บอร์ดพกพา	8
1.6 เม้าส์แบบทั่วไป	8
1.7 เม้าส์แบบแสง	9
1.8 แทรคบอล	9
1.9 แผ่นรองสัมผัส	9
1.10 จอยสติ๊ก	10
1.11 จอยสติ๊กหรือทัชสกรีน	10
1.12 ปากกาแสง	10
1.13 สไตลัส	11
1.14 ดิจิไทเซอร์	11
1.15 ไมโครโฟน	11
1.16 กล้องถ่ายภาพดิจิทัล	12
1.17 กล้องถ่ายวิดีโอดิจิทัล	12
1.18 เว็บแคม	13
1.19 สแกนเนอร์	13
1.20 โอเอ็มอาร์	13
1.21 เครื่องอ่านบาร์โค้ด	14
1.22 เอ็มไอซีอาร์	14
1.23 ไบโอมेटริกส์	15
1.24 ซีพียู	15
1.25 เอสดีแรม	16
1.26 ดีดีอาร์ เอสดีแรม	16
1.27 ดีดีอาร์-ทู เอสดีแรม	17
1.28 เมนบอร์ด	17
1.29 ฮาร์ดดิสก์	18
1.30 พายท์ออฟเซลเทอร์มินัล	18
1.31 จอซีอาร์ที	19
1.32 จอแอลซีดี	19
1.33 โพรเจคเตอร์	20
1.34 เครื่องพิมพ์แบบดอทเมทริกซ์	20

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
1.35 เครื่องพิมพ์แบบเลเซอร์	20
1.36 เครื่องพิมพ์แบบพ่นหมึก	21
1.37 ลำโพง	21
1.38 หูฟัง	22
1.39 การติดตั้งวีชวลสตูดิโออัลทิเมทแบบอัตโนมัติ	27
1.40 การติดตั้งวีชวลสตูดิโออัลทิเมทแบบติดตั้งเอง	28
1.41 ตำแหน่งเลือกติดตั้งไมโครซอฟท์วีชวลสตูดิโอ	28
1.42 ตำแหน่งปุ่ม Next เพื่อดำเนินการติดตั้ง	29
1.43 ตำแหน่งเลือกเครื่องหมายเพื่อยอมรับข้อเสนอ	29
1.44 ตำแหน่งเลือกรูปแบบการติดตั้งแบบเต็มรูปแบบ	30
1.45 โปรแกรมวีชวลสตูดิโอกำลังติดตั้ง	30
1.46 หน้าจอเมื่อติดตั้งโปรแกรมสำเร็จ	31
2.1 รูปแบบผังงานแบบตามลำดับ	40
2.2 ผังงานแบบเลือกเส้นทางรูปแบบที่ 1	42
2.3 ผังงานแบบเลือกเส้นทางรูปแบบที่ 2	43
2.4 ผังงานแบบเลือกเส้นทางรูปแบบที่ 3	44
2.5 ผังงานแบบทำซ้ำรูปแบบที่ 1	51
2.6 ผังงานแบบทำซ้ำรูปแบบที่ 2	52
3.1 เทคนิคการใช้แสงในการออกแบบปุ่มบนเว็บไซต์หรือแอปพลิเคชัน	61
3.2 การใช้แสงในการออกแบบรอยต่อ	62
3.3 การออกแบบปุ่มการเชื่อมต่อกับผู้ใช้แบบเรียบง่าย	63
3.4 โปรแกรมที่ถูกออกแบบแบบกึ่งเรียบ	64
3.5 การออกแบบวัตถุที่ใช้แสงเงาเพิ่มความลึก	64
3.6 การใช้แสงเงาทำให้เกิดความลึก ความนูน	64
3.7 ตัวอย่างการออกแบบโทนสีขาว - ดำ	65
3.8 ตัวอย่างการออกแบบโทนสีขาว - ดำ ที่เพิ่มสีเข้าไปอีก 1 สี	66
3.9 ตัวอย่างการเปลี่ยนความเข้มข้นและความสว่างของโทนสีโดยที่เนื้อสีคงที่	67
3.10 ตัวอย่างการออกแบบนาฬิกาจับเวลา	67
3.11 ตัวอย่างโปรแกรมเล่นเพลง	68
3.12 ตัวอย่างการใช้สีดำโปร่งใสประมาณ 30% เพื่อให้ตัวหนังสืออ่านง่ายขึ้น	69
3.13 ตัวอย่างงานออกแบบโปรแกรมแบบใส่ตัวอักษรไว้ในกล่อง	69
3.14 ตัวอย่างงานเบลอรูปร่างพื้นหลัง	70
3.15 ตัวอย่างการออกแบบให้พื้นผิวด้านล่างเลื่อนราง	70
3.16 ตัวอย่างชุดแบบอักษรไทย แบบมีหัวและไม่มีหัว	71

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.17 ตัวอย่างการเพิ่มความเด่นของตัวหนังสือ	71
3.18 ตัวอย่างการลดความเด่นของตัวหนังสือ	72
3.19 ตัวอย่างการออกแบบที่ใช้เทคนิคเด่นขึ้นและเด่นน้อยลงในจุดเดียวกัน	72
4.1 การเข้าสู่โปรแกรมไมโครซอฟต์วิซวลสตูดิโอ 2010	79
4.2 การสร้างโครงการใหม่	80
4.3 หน้าต่างของโครงการ	80
4.4 คอนโทรลปุ่มกด	81
4.5 หน้าต่างสำหรับการเขียนคำสั่ง	82
4.6 การเขียนโค้ดลงที่ปุ่มปุ่มกด	82
4.7 ตัวอย่างการทดสอบเขียนโปรแกรม	83
4.8 คอนโทรลโอเวิลเซพ	84
4.9 การสร้างปุ่มกด ควบคุมหลอดไฟ ติด - ดับ	84
4.10 ผลลัพธ์การสร้างปุ่มกด ควบคุมหลอดไฟ ติด - ดับ	86
4.11 คอนโทรลฟิเจอร์บ็อกซ์	86
4.12 การสร้างปุ่มกด ควบคุมมองเห็น - มองไม่เห็น	87
4.13 การเลือกรูปภาพลงคอนโทรลฟิเจอร์บ็อกซ์	87
4.14 ผลลัพธ์การสร้างปุ่มกดควบคุมมองเห็น - มองไม่เห็น	88
4.15 คอนโทรลลาเบล	89
4.16 การสร้างปุ่มกด ควบคุมการแสดงข้อความ	90
4.17 ผลลัพธ์การสร้างปุ่มกดควบคุมการแสดงข้อความ	91
4.18 คอนโทรลเท็กซ์บ็อกซ์	91
4.19 การสร้างคอนโทรลเท็กซ์บ็อกซ์สำหรับการป้อนข้อมูลและข้อความ	92
4.20 ผลลัพธ์การแสดงผลการสร้างคอนโทรลเท็กซ์บ็อกซ์สำหรับการป้อนข้อมูลและข้อความ	93
4.21 คอนโทรลเช็คบ็อกซ์	94
4.22 คอนโทรลเรดิโอบัตตอน	94
4.23 คอนโทรลกรุปบ็อกซ์	95
4.24 คอนโทรลสโครลบาร์	95
4.25 การสร้างและใช้งานคอนโทรลสโครลบาร์	96
4.26 ผลลัพธ์การสร้างและใช้งานคอนโทรลสโครลบาร์	97
4.27 การสร้างและใช้งานวัตถุเคลื่อนที่	97
4.28 ค่าคุณสมบัติของโอเวิลเซพ	98
4.29 ผลลัพธ์การสร้างและใช้งานวัตถุเคลื่อนที่	99
4.30 คอนโทรลไทมเมอร์	99
4.31 การเขียนโปรแกรมใช้งานคอนโทรลไทมเมอร์	100

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

4.32	ผลลัพธ์การใช้งานคอนโทรลไทมเมอร์	100
4.33	ตัวอย่างอุปกรณ์ช่วยนำทาง	101
4.34	ตัวอย่างหน้าจอเมนูสินค้าที่สามารถใส่ภาพสินค้าได้เอง	102
4.35	ตัวอย่างหน้าจอสถานะโต๊ะ สำหรับร้านอาหาร	102
4.36	ตัวอย่างคอนโทรลปุ่มกดแสดงข้อความ “Test...”	104
4.37	ตัวอย่างคอนโทรลปุ่มกดเมื่อถูกกดแล้วกลายเป็นสีเหลือง	104
4.38	ตัวอย่างคอนโทรลไอเวิลเซพแสดงผลภายในเป็นเส้นขีดในแนวนอน	104
4.39	ตัวอย่างเส้นขอบคอนโทรลไอเวิลเซพที่มีความหนา	105
4.40	ตัวอย่างขนาดของคอนโทรลลาเบลมีขนาดใหญ่ขึ้น	105
5.1	ทดลอง ชนิดของข้อมูล ตัวแปรและค่าคงที่	114
5.2	การแจ้งเตือนการใส่ข้อมูลเกินหรือผิดจากชนิดของข้อมูล	114
5.3	ผลลัพธ์โอเพอร์เรเตอร์ทางคณิตศาสตร์	118
5.4	โอเพอร์เรเตอร์สำหรับการเชื่อมต่อสตริง	118
6.1	โปรแกรมคำนวณหาพื้นที่วงกลม	126
6.2	ผลลัพธ์โปรแกรมคำนวณหาพื้นที่วงกลม	126
6.3	โปรแกรมคำนวณหาเส้นรอบรูปสี่เหลี่ยมผืนผ้า	127
6.4	ผลลัพธ์โปรแกรมคำนวณหาเส้นรอบรูปสี่เหลี่ยมผืนผ้า	128
6.5	ผังงานแบบเลือกเส้นทางด้วยคำสั่ง If ... Then แบบที่ 1	129
6.6	หน้าจอโปรแกรมการลดราคาเมื่อสินค้าเกิน 100 บาท	130
6.7	ผลลัพธ์โปรแกรมการลดราคาเมื่อสินค้าเกิน 100 บาท	131
6.8	ผังงานแบบเลือกเส้นทางด้วยคำสั่ง If ... Then แบบที่ 2	131
6.9	หน้าจอโปรแกรมปุ่มสำหรับกด	133
6.10	ผลลัพธ์โปรแกรมปุ่มสำหรับกด	134
6.11	หน้าจอโปรแกรมการสร้างการป้อนรหัสผ่าน	135
6.12	ผลลัพธ์โปรแกรมการสร้างการป้อนรหัสผ่าน	135
6.13	ผังงานแบบเลือกเส้นทางด้วยคำสั่ง If ... Then แบบที่ 3	136
6.14	หน้าจอโปรแกรมการประเมินผลงาน	137
6.15	ผลลัพธ์โปรแกรมการประเมินผลงาน	138
6.16	ผังงานแบบเลือกเส้นทางด้วยคำสั่ง Select ... Case	138
6.17	หน้าจอการรับค่าตัวเลขและแสดงข้อความ	140
6.18	ผลลัพธ์โปรแกรมการรับค่าตัวเลขและแสดงข้อความ	141
6.19	ผังงานแบบวนซ้ำด้วยคำสั่ง For ... Next	141
6.20	หน้าจอโปรแกรมการพิมพ์เลข 1-10 ลงบนคอนโทรลลิทบ็อกซ์	143
6.21	ผลลัพธ์หน้าจอโปรแกรมการพิมพ์เลข 1-10 ลงบนคอนโทรลลิทบ็อกซ์	144

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
6.22 หน้าจอการพิมพ์เลขตามข้อมูล จุดเริ่มต้น จุดสิ้นสุดและค่าของจำนวนนับที่เพิ่มขึ้น ผ่านทางคอนโทรลเท็กซ์บ็อกซ์ลงบนคอนโทรลลิทบ็อกซ์	145
6.23 ผลลัพธ์การพิมพ์เลขตามข้อมูล จุดเริ่มต้น จุดสิ้นสุดและค่าของจำนวนนับที่เพิ่มขึ้น ผ่านทางคอนโทรลเท็กซ์บ็อกซ์ลงบนคอนโทรลลิทบ็อกซ์	146
6.24 ผังงานแบบวนซ้ำด้วยคำสั่ง Do While ... Loop	146
6.25 หน้าจอการพิมพ์เลข 1-10 ลงบนคอนโทรลลิทบ็อกซ์โดยนับเพิ่มขึ้นทีละ 2	148
6.26 ผลลัพธ์การพิมพ์เลข 1-10 ลงบนคอนโทรลลิทบ็อกซ์โดยนับเพิ่มขึ้นทีละ 2	149
6.27 ผังงานแบบวนซ้ำด้วยคำสั่ง Do ... Loop While	149
6.28 หน้าจอโปรแกรมการพิมพ์เลข 1-30 ลงบนคอนโทรลลิทบ็อกซ์	151
6.29 ผลลัพธ์โปรแกรมการพิมพ์เลข 1-30 ลงบนคอนโทรลลิทบ็อกซ์	152
6.30 หน้าจอโปรแกรมการพิมพ์สูตรคูณลงบนคอนโทรลลิทบ็อกซ์	153
6.31 ผลลัพธ์โปรแกรมการพิมพ์สูตรคูณลงบนคอนโทรลลิทบ็อกซ์	154
6.32 ผังงานแบบวนซ้ำด้วยคำสั่ง Do Until ... Loop	155
6.33 หน้าจอโปรแกรมการพิมพ์เลข 20-1 ลงบนคอนโทรลลิทบ็อกซ์	156
6.34 ผลลัพธ์โปรแกรมการพิมพ์เลข 20-1 ลงบนคอนโทรลลิทบ็อกซ์	157
6.35 ผังงานแบบวนซ้ำด้วยคำสั่ง Do ... Loop Until	158
6.36 หน้าจอโปรแกรมการพิมพ์เลข 50-1 ลงบนคอนโทรลลิทบ็อกซ์ โดยลดลงทีละ 5	159
6.37 ผลลัพธ์โปรแกรมการพิมพ์เลข 50-1 ลงบนคอนโทรลลิทบ็อกซ์ โดยลดลงทีละ 5	160
7.1 โปรแกรมเครื่องคิดเลข	167
7.2 การแสดงผลโปรแกรมเครื่องคิดเลข	171
7.3 โปรแกรมตั้งเวลาปลุก	171
7.4 การแสดงผลโปรแกรมตั้งเวลาปลุก	173
7.5 โปรแกรมแสดงตัวอักษรวิ่ง	173
7.6 การแสดงผลโปรแกรมแสดงตัวอักษรวิ่ง	174
7.7 โปรแกรมบอร์ดควบคุมระดับน้ำ	174
7.8 การแสดงผลโปรแกรมบอร์ดควบคุมระดับน้ำ	177
7.9 โปรแกรมบอร์ดควบคุมไฟในถังผสม	178
7.10 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมที่ปุ่ม Mixer	183
7.11 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมที่ปุ่ม Heater	183
7.12 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมเมื่อกดปุ่ม Inlet Valve	183
7.13 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมเมื่อกดปุ่ม Outlet Valve	184
7.14 โปรแกรมควบคุมไฟจราจร	184
7.15 การแสดงผลโปรแกรมควบคุมไฟจราจร	192

สารบัญตาราง

ตารางที่	หน้า
5.1 การระบุประเภทและชนิดของข้อมูล	111
5.2 ตัวเลขจำนวนเต็ม	111
5.3 ตัวเลขทศนิยม	112
5.4 ตัวเลขจำนวนเต็มและตัวเลขทศนิยม	112
5.5 ตัวอักษร	112
5.6 โอเปอเรเตอร์ทางคณิตศาสตร์	115
5.7 โอเปอเรเตอร์สำหรับการเปรียบเทียบ	115
5.8 โอเปอเรเตอร์สำหรับการกำหนดค่า	115
5.9 โอเปอเรเตอร์ทางตรรกะ	116
5.10 ลำดับในการทำงานของโอเปอเรเตอร์	117

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

แผนบริหารการสอนประจำรายวิชา

รายวิชา การโปรแกรมคอมพิวเตอร์
รวม เรียน 15 สัปดาห์

หน่วยกิต 3(2-3-5)
รวม 75 ชั่วโมง

คำอธิบายรายวิชา

หลักการของคอมพิวเตอร์ อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ การทำงานร่วมกันของคอมพิวเตอร์ หลักการประมวลผลข้อมูลด้วยคอมพิวเตอร์ ลักษณะข้อมูลที่ใช้ในคอมพิวเตอร์ ข้อมูลชนิดโครงสร้างไฟล์ การกำหนดชนิดของข้อมูลและตัวแปรต่าง ๆ การเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาคอมพิวเตอร์ระดับสูง การออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรม

วัตถุประสงค์ทั่วไป

เพื่อให้นักศึกษาเข้าใจระบบคอมพิวเตอร์ และสามารถเขียนผังงาน และเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาคอมพิวเตอร์ระดับสูงสำหรับการออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรมเบื้องต้นได้

เนื้อหา

75 ชั่วโมง

บทที่ 1 คอมพิวเตอร์เบื้องต้น

5 ชั่วโมง

1. หลักการของคอมพิวเตอร์
2. ลักษณะข้อมูลที่ใช้ในคอมพิวเตอร์และข้อมูลชนิดโครงสร้างไฟล์
3. อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์
4. โปรแกรมวิซวลเบสิก
5. การติดตั้งโปรแกรมวิซวลเบสิกจากไมโครซอฟท์วิซวลสตูดิโอ

บทที่ 2 การเขียนผังงาน

10 ชั่วโมง

1. หลักการเขียนผังงานเบื้องต้น
2. การเขียนผังงานแบบลำดับ
3. การเขียนผังงานแบบเลือกเส้นทาง
4. การเขียนผังงานแบบทำซ้ำ

บทที่ 3 การออกแบบหน้าจอ

5 ชั่วโมง

1. การใช้แสง
2. การใช้สีขาว - ดำในการออกแบบ
3. การมีพื้นที่ว่าง
4. การวางตัวหนังสือบนรูปภาพ
5. การเพิ่มหรือลด ความเด่นของตัวหนังสือ
6. การเลือกใช้ชุดแบบอักษรให้เหมาะสม

บทที่ 4 การใช้งานคอนโทรลเบื้องต้น

10 ชั่วโมง

1. การสร้างโครงงานใหม่
2. คอนโทรลบัตทอน
3. คอนโทรลโอเวิลเซพ
4. คอนโทรลฟิกเจอร์บ็อกซ์
5. คอนโทรลลาเบล
6. คอนโทรลเท็กซ์บ็อกซ์
7. คอนโทรลเช็คบ็อกซ์
8. คอนโทรลเรดิโอบัตทอน
9. คอนโทรลกรุปบ็อกซ์
10. คอนโทรลสโครลบาร์
11. คอนโทรลไทมเมอร์
12. การตั้งชื่อคอนโทรลและอ็อบเจ็กต์ตามคำแนะนำของไมโครซอฟท์

บทที่ 5 โครงสร้างทางภาษา

10 ชั่วโมง

1. ตัวแปร
2. คำสงวน
3. ค่าคงที่
4. ชนิดของข้อมูล
5. การกำหนดค่าให้ตัวแปร
6. โอเปอเรเตอร์
7. ลำดับการทำงานของโอเปอเรเตอร์
8. การเขียนคำอธิบายโปรแกรม
9. คำสั่งในการเขียนโปรแกรม
10. การรับข้อมูลและแสดงข้อความ

บทที่ 6 การเขียนแอปพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์

15 ชั่วโมง

1. คำสั่งควบคุมการทำงานแบบลำดับ
2. คำสั่งควบคุมการทำงานแบบเลือกเส้นทาง
3. คำสั่งควบคุมการทำงานแบบทำซ้ำ

บทที่ 7 การออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรม

20 ชั่วโมง

1. โปรแกรมเครื่องคิดเลข
2. โปรแกรมตั้งเวลาปลุก
3. โปรแกรมแสดงตัวอักษรวิ่ง
4. โปรแกรมบอร์ตควบคุมระดับน้ำ
5. โปรแกรมบอร์ตควบคุมไฟในถังผสม
6. โปรแกรมควบคุมไฟจราจร

กิจกรรมการเรียนการสอน

- 1 บรรยายคำสั่ง ขั้นตอน วิธีปฏิบัติ
- 2 มอบหมายงานให้ทดลองปฏิบัติ

สื่อการเรียนการสอน

- 1 เอกสารประกอบการสอน
- 2 ใบงานทดลอง
- 3 โปรแกรมไมโครซอฟท์วิซวลสตูดิโอ
- 4 เครื่องฉายโปรเจคเตอร์

การวัดและประเมินผล

การวัดผล

- 1 คะแนนระหว่างภาคเรียน 60 คะแนน
- 2 คะแนนสอบปลายภาคเรียน 40 คะแนน

การประเมินผล

- 100 – 80 ได้ระดับ A
79 – 75 ได้ระดับ B+
74 – 70 ได้ระดับ B
69 – 65 ได้ระดับ C+
64 – 60 ได้ระดับ C
59 – 55 ได้ระดับ D+
54 – 50 ได้ระดับ D
ต่ำกว่า 50 ได้ระดับ F

แผนบริหารการสอนประจำบทที่ 1

หัวข้อเนื้อหาบทที่ 1

1. หลักการของคอมพิวเตอร์
2. ลักษณะข้อมูลที่ใช้ในคอมพิวเตอร์และข้อมูลชนิดโครงสร้างไฟล์
3. อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์
4. โปรแกรมวิซวลเบสิก
5. การติดตั้งโปรแกรมวิซวลเบสิกจากไมโครซอฟท์วิซวลสตูดิโอ
6. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 1 แล้วนักศึกษาสามารถ

1. รู้หลักการของคอมพิวเตอร์เบื้องต้น
2. รู้จักลักษณะข้อมูลที่ใช้ในคอมพิวเตอร์และข้อมูลชนิดโครงสร้างไฟล์
3. รู้ส่วนประกอบ อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ของคอมพิวเตอร์
4. รู้จักโปรแกรมวิซวลเบสิก
5. รู้วิธีการติดตั้งโปรแกรมวิซวลเบสิก

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายข้อตกลงในการเรียนการสอน และเนื้อหารายวิชา
2. ให้นักศึกษาได้ศึกษาเรียนรู้ ผ่านการบรรยายจากเอกสารประกอบการสอน
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 1

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
4. แบบฝึกหัดบทที่ 1

การวัดผลและประเมินผล

1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากกิจกรรมร่วม
3. สังเกตจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด

บทที่ 1

คอมพิวเตอร์เบื้องต้น

คอมพิวเตอร์ เป็นอุปกรณ์อิเล็กทรอนิกส์ที่เข้ามามีบทบาทในชีวิตประจำวัน เนื่องจากความสามารถที่เป็นเหมือนเครื่องคำนวณอิเล็กทรอนิกส์ที่มีการทำงานแบบอัตโนมัติ ทำหน้าที่เหมือนสมองกล ที่คอยช่วยแก้ไขปัญหาง่าย ๆ ไปจนถึงซับซ้อนตามคำสั่งของโปรแกรม โดยขั้นตอนการทำงานของคอมพิวเตอร์ประกอบด้วย การรับโปรแกรมและข้อมูล ซึ่งโปรแกรมในที่นี้หมายถึงชุดของคำสั่ง ที่สั่งให้คอมพิวเตอร์ทำงาน หรือเรียกว่าโปรแกรมคอมพิวเตอร์ ส่วนข้อมูลนั้น ได้แก่ ตัวเลขหรือตัวอักษร ที่ต้องการให้คอมพิวเตอร์ประมวลผล ซึ่งการประมวลผลนั้นคือการจัดระเบียบแบบแผนของข้อมูลหรือชุดข้อมูล ตลอดจนคำนวณ เปรียบเทียบ วิเคราะห์โดยใช้สูตรทางคณิตศาสตร์ และหลังจากที่ข้อมูลได้ผ่านการประมวลผลเป็นที่เรียบร้อยแล้วผลลัพธ์ก็จะถูกนำออกไปในรูปแบบต่าง ๆ ที่ผู้ใช้สามารถนำไปใช้ประโยชน์ได้ ซึ่งเนื้อหาในบทนี้ได้อธิบายถึงหลักการตั้งที่กล่าวไว้ ตลอดจนอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ การทำงานร่วมกันของคอมพิวเตอร์ และการลงโปรแกรมไมโครซอฟท์วิซวลสตูดิโอ (Microsoft visual studio) เพื่อใช้สำหรับการเรียนการสอนในบทต่อไป

หลักการของคอมพิวเตอร์

คอมพิวเตอร์เป็นอุปกรณ์ที่ทำงานเป็นระบบจึงต้องมีการแบ่งส่วนประกอบออกเป็น ส่วน ๆ ซึ่งแต่ละส่วนนั้นต้องทำงานประสานกันเพื่อบรรลุวัตถุประสงค์เดียวกัน ดังนั้นระบบที่สมบูรณ์แบบจึงประกอบด้วยส่วนประกอบสำคัญ 4 ส่วนดังนี้ (โอภาส, 2547)

1. หน่วยรับข้อมูล (Input unit) เป็นระบบหนึ่งที่ช่วยให้คอมพิวเตอร์สามารถโต้ตอบกับผู้ใช้งานได้ โดยทำหน้าที่รับข้อมูลเข้าสู่เครื่องคอมพิวเตอร์ ผ่านอุปกรณ์ เช่น เมาส์สำหรับกดเลือกสิ่งงานสิ่งต่าง ๆ ในโปรแกรม แป้นพิมพ์สำหรับพิมพ์ตัวอักษร เครื่องสแกนเนอร์สำหรับสแกนรูปภาพ ไมโครโฟนสำหรับพูดอัดเสียง หรือกล้องวิดีโอที่รับภาพเคลื่อนไหวไปใช้ในคอมพิวเตอร์ เป็นต้น โดยข้อมูลที่รับจะถูกบันทึกไว้ในฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ เพื่อบรรณาการนำไปประมวลผลหรือแสดงผลในลำดับต่อไป

2. หน่วยประมวลผลกลาง (CPU: Central processing unit) หรือไมโครโปรเซสเซอร์ ทำหน้าที่ในการคำนวณและประมวลผล ซึ่งแบ่งข้อมูลย่อยออกเป็น 2 หน่วยคือ

- 2.1 หน่วยควบคุม ทำหน้าที่ในการควบคุมลำดับขั้นตอนของการประมวลผลของอุปกรณ์ต่าง ๆ ตลอดจนประสานงานระหว่างหน่วยประมวลผลกลาง หน่วยรับข้อมูล หน่วยแสดงผล และหน่วยความจำ

- 2.2 หน่วยคำนวณและตรรก ทำหน้าที่เปรียบเทียบข้อมูลจากหน่วยควบคุมและหน่วยความจำ

3. หน่วยแสดงผล (Output unit) หลังจากทีคอมพิวเตอร์ได้ประมวลผลเป็นที่เรียบร้อยแล้ว ก็แสดงผลข้อมูลในรูปแบบต่าง ๆ อาทิ ข้อความ ภาพนิ่ง ที่แสดงผลในหน้าจอคอมพิวเตอร์ หรือเครื่องพิมพ์ ตลอดจนเสียงที่แสดงออกทางลำโพง เป็นต้น

4. หน่วยความจำ (Memory unit) ทำหน้าที่ในการเก็บข้อมูลให้กับคอมพิวเตอร์ โดยหน่วยความจำมีอยู่ 2 ชนิดคือ หน่วยความจำถาวรและหน่วยความจำชั่วคราว

4.1 หน่วยความจำถาวร เป็นหน่วยความจำที่จำข้อมูลได้ตลอดเวลา ซึ่งสามารถแบ่งย่อยได้เป็นหน่วยความจำแบบพรอม (PROM: Programmable read only memory) เป็นหน่วยความจำที่บันทึกโปรแกรมได้เพียงครั้งเดียว และหน่วยความจำแบบอีพรอม (EPROM: Erasable programmable read only memory) เป็นหน่วยความจำที่สามารถโปรแกรมและลบข้อมูลได้หลาย ๆ ครั้ง โดยหน่วยความจำแบบถาวรหากไม่จ่ายไฟเข้าหน่วยความจำ ข้อมูลในหน่วยความจำจะยังคงอยู่ไม่สูญหาย

4.2 หน่วยความจำชั่วคราว จะต่างจากหน่วยความจำถาวรที่ หน่วยความจำนี้จะจำข้อมูลได้เฉพาะช่วงที่เปิดไฟเข้าเครื่องคอมพิวเตอร์เท่านั้น เมื่อขาดไฟเลี้ยงข้อมูลจะสูญหายทันที

นอกจากหน่วยความจำถาวรและหน่วยความจำชั่วคราวแล้ว ยังมีหน่วยเก็บข้อมูลสำรอง (Secondary storage) ที่ทำหน้าที่เก็บข้อมูลเพื่อนำไปใช้ในโอกาสต่อไป โดยมีอยู่ด้วยกันหลายชนิด ได้แก่ ฮาร์ดดิสก์ ดิสก์ไดรฟ์ ซีดีรอม ดีวีดีรอม แฟลชไดรฟ์ เป็นต้น

จากส่วนประกอบสำคัญ 4 ส่วน สามารถสรุปหลักการของคอมพิวเตอร์เบื้องต้นได้ดังนี้คือ เริ่มจากผู้ใช้งานป้อนข้อมูลผ่านอุปกรณ์ของหน่วยรับข้อมูล เช่น คีย์บอร์ด เมาส์ จากนั้นข้อมูลจึงถูกเปลี่ยนเป็นสัญญาณดิจิทัล แล้วส่งต่อไปยังหน่วยประมวลผลกลางที่คอยทำหน้าที่ควบคุมและคำนวณต่าง ๆ ตามความเร็วของหน่วยประมวลผลกลาง เพื่อนำไปใช้งานหรือส่งไปยังหน่วยแสดงผล เช่น จอภาพ เครื่องพิมพ์ ฯลฯ เพื่อให้แสดงผลตามความต้องการของผู้ใช้ โดยมีหน่วยความจำในแบบต่าง ๆ ไว้ช่วยบันทึกข้อมูลตามความต้องการหรือตามลักษณะการใช้งานทั้งในส่วนของผู้ใช้และคอมพิวเตอร์

ลักษณะข้อมูลที่ใช้ในคอมพิวเตอร์และข้อมูลชนิดโครงสร้างไฟล์

สิ่งสำคัญอย่างหนึ่งที่เป็นองค์ประกอบในระบบคอมพิวเตอร์ได้แก่ ข้อมูล เพราะเป็นสิ่งที่ผู้ใช้หรือนักออกแบบโปรแกรมเขียนขึ้นเพื่อผลิตผลลัพธ์ที่ต้องการออกมา โดยข้อมูลที่สามารถนำมาใช้กับคอมพิวเตอร์ได้ มี 5 ประเภท (ณัฐกร สงคราม, ม.ป.ป.) ได้แก่

1 ข้อมูลแบบรูปแบบ (Formatted data) เป็นข้อมูลที่มีรูปแบบที่แน่นอน จำพวกอักขระทั้งนี้อาจหมายถึงตัวอักษร ตัวเลข โดยข้อมูลอาจเก็บในรูปของรหัส เมื่อโปรแกรมอ่านข้อมูลอาจจะต้องนำรหัสนั้นมาตีความหมายเป็นต้น

2 ข้อมูลแบบข้อความ (Text) เป็นข้อมูลที่ไม่มีการจัดรูปแบบที่แน่นอน อาจเป็นได้ทั้งอักขระในแบบข้อความ ตัวอักษร ตัวเลข สมการ แต่ไม่รวมภาพต่าง ๆ ลักษณะข้อมูลแบบนี้ไม่ต้องนำข้อมูลที่เก็บมาตีความหมายอีก

3 ข้อมูลแบบภาพลักษณ์ (Images) เป็นข้อมูลที่ถูกสร้างขึ้น อาจเป็นภาพกราฟ รูปภาพ หรือภาพวาด ซึ่งคอมพิวเตอร์สามารถเก็บภาพ ปรับขยายภาพและเคลื่อนย้ายภาพเหล่านั้นได้เหมือนกับข้อมูลแบบข้อความ

4 ข้อมูลแบบเสียง (Audio) เป็นข้อมูลเสียง ที่มีลักษณะของการจัดเก็บเหมือนข้อมูลแบบภาพ

5 ข้อมูลแบบภาพและเสียง (Video) เป็นข้อมูลเสียงและภาพ ที่ถูกผสมผสานรูปภาพและเสียงเข้าด้วยกัน ลักษณะของข้อมูล

โดยการนำข้อมูลไปใช้นั้น มีระดับข้อมูลชนิดโครงสร้างไฟล์ดังนี้

1. บิต (Bit) คือ ข้อมูลที่มีขนาดเล็กที่สุด ได้แก่ เลข 0 หรือ เลข 1 เท่านั้น
2. ไบต์ (Byte) หรือ อักขระ (Character) ได้แก่ ตัวเลข หรือ ตัวอักษร หรือ สัญลักษณ์พิเศษ 1 ตัว เช่น 0, 1, ..., 9, A, B, ..., Z และเครื่องหมายต่าง ๆ ซึ่ง 1 ไบต์เท่ากับ 8 บิต หรือ ตัวอักษร 1 ตัว เป็นต้น
3. ฟیلด์ (Field) ได้แก่ ไบต์ หรือ อักขระตั้งแต่ 1 ตัวขึ้นไป รวมกันเป็นฟیلด์ เช่น เลขประจำตัวพนักงาน เป็นต้น
4. เรคคอร์ด (Record) ได้แก่ ฟیلด์ตั้งแต่ 1 ฟیلด์ ขึ้นไป ที่มีความสัมพันธ์เกี่ยวข้องรวมกันเป็นเรคคอร์ด เช่น ชื่อ นามสกุล เลขประจำตัว ยอดขาย ข้อมูลของพนักงาน 1 คน เป็น 1 เรคคอร์ด
5. ไฟล์ (Files) หรือ แฟ้มข้อมูลได้แก่ เรคคอร์ดหลาย ๆ เรคคอร์ดรวมกัน ซึ่งเป็นเรื่องเดียวกัน เช่น ข้อมูลของประวัติพนักงานแต่ละคนรวมกันทั้งหมดเป็นไฟล์หรือแฟ้มข้อมูลเกี่ยวกับประวัติพนักงานของบริษัท เป็นต้น
6. ฐานข้อมูล (Database) คือ การเก็บรวบรวมไฟล์ข้อมูลหลาย ๆ ไฟล์ที่เกี่ยวข้องกันมา รวมเข้าด้วยกันเช่น ไฟล์ข้อมูลของแผนกต่าง ๆ มารวมกันเป็นฐานข้อมูลของบริษัท เป็นต้น

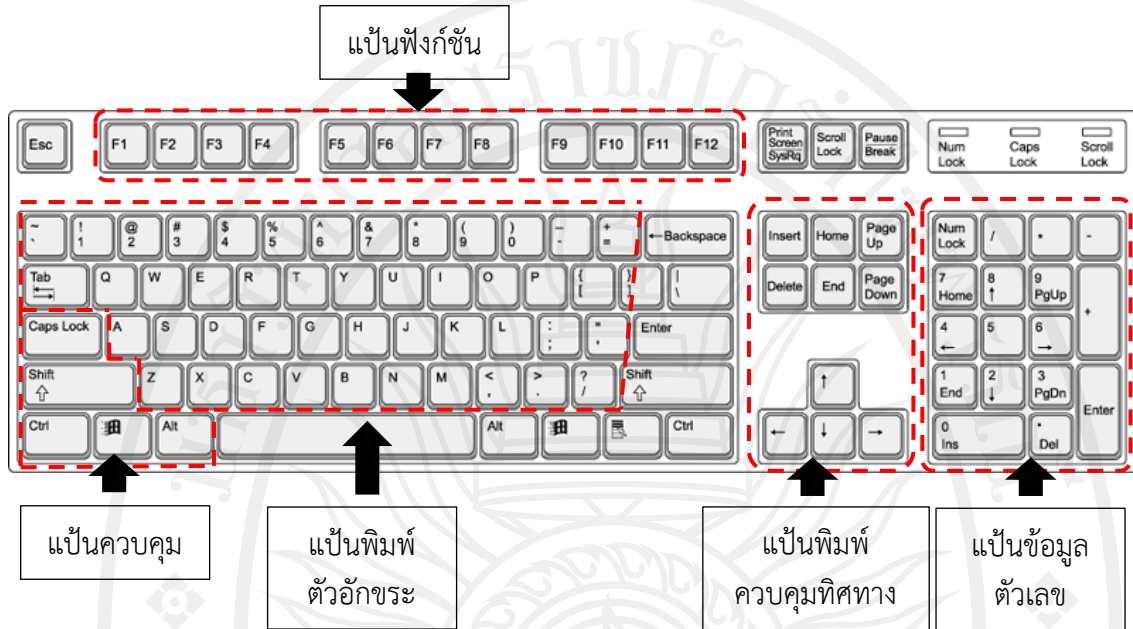
อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์

1. ฮาร์ดแวร์ (Hardware) หมายถึง อุปกรณ์ที่เกี่ยวข้องกับการใช้งานร่วมกับคอมพิวเตอร์ เช่น จอภาพ แป้นพิมพ์ เมาส์ เครื่องพิมพ์ เป็นต้น โดยมีรูปร่างลักษณะที่สามารถมองเห็นและสัมผัสได้ และอยู่ทั้งในตัวเครื่องและนอกตัวเครื่อง บางครั้งนิยมเรียกว่าดีไวซ์ (Device) ที่ทำงานประสานกันตั้งแต่หน่วยรับข้อมูลไปจนถึงหน่วยแสดงผล ซึ่งอุปกรณ์ฮาร์ดแวร์สามารถแบ่งได้ดังนี้ (วิโรจน์และสุพรรณษา, 2552)

1.1 อุปกรณ์ฮาร์ดแวร์ของหน่วยรับข้อมูล เป็นอุปกรณ์ที่เกี่ยวข้องกับการนำข้อมูลมายังระบบคอมพิวเตอร์เพื่อประมวลผลโดยอาจมาเป็นตัวเลข ภาพกราฟิก เสียง หรือวิดีโอ เป็นต้น ซึ่งแบ่งเป็นประเภทตามที่พบเห็นในปัจจุบันได้ดังนี้

1.1.1 ประเภทปุ่มกด (Keyed device) หรืออุปกรณ์ประเภทคีย์บอร์ด (Keyboard) จัดเป็นอุปกรณ์นำข้อมูลเข้าที่นิยมใช้กันมากที่สุดในปัจจุบัน กลุ่มของแป้นพิมพ์คีย์บอร์ดวางเรียงต่อกันเหมือนกับเครื่องพิมพ์ดีด โดยสามารถรับข้อมูลป้อนเข้าที่เป็นตัวอักษร อักขระพิเศษ ตัวเลข รวมถึงชุดคำสั่งต่าง ๆ ที่ผู้ใช้สามารถเข้าใจ แล้วส่งข้อมูลที่ป้อนไปยังหน่วยความจำของระบบและแปลงให้เป็นรหัสที่คอมพิวเตอร์เข้าใจ

คีย์บอร์ดมาตรฐาน (Standard keyboard) เป็นคีย์บอร์ดที่นิยมใช้กันโดยทั่วไป ดังภาพที่ 1.1 ซึ่งประกอบไปด้วยกลุ่มของแป้นพิมพ์หลัก ๆ ดังนี้



ภาพที่ 1.1 กลุ่มของแป้นพิมพ์บนคีย์บอร์ดมาตรฐาน ที่มา ดัดแปลงจาก (ศุภรณ์ภา. ม.ป.ป.)

- แป้นฟังก์ชัน : เป็นกลุ่มของแป้นพิมพ์ในการเลือกคำสั่งพิเศษหรือคำสั่งลัด
- แป้นควบคุม : เป็นกลุ่มของแป้นพิมพ์ที่ใช้ทำงานร่วมกับแป้นพิมพ์อื่น ๆ เช่น แป้น Ctrl, แป้น Shift หรือ แป้น Alt เป็นต้น
- แป้นพิมพ์ตัวอักษร : กลุ่มของแป้นพิมพ์นี้จะมีบริเวณใหญ่ที่สุด ประกอบไปด้วยแผงอักขระสำหรับให้ผู้ใช้ สื่อสารกับคอมพิวเตอร์เป็นตัวอักษร
- แป้นพิมพ์ควบคุมทิศทาง : เป็นกลุ่มของแป้นพิมพ์สำหรับเปลี่ยนตำแหน่งการทำงานของเคอร์เซอร์
- แป้นข้อมูลตัวเลข : เป็นกลุ่มของแป้นพิมพ์ที่ช่วยป้อนข้อมูลที่เป็นตัวเลข ให้ใช้งานได้ง่ายขึ้น

คีย์บอร์ดติดตั้งภายใน (Built-in keyboard) เป็นคีย์บอร์ดที่ติดตั้งแป้นข้อมูลตัวเลขออก มีการย่อขนาดตำแหน่งของแป้นให้เล็กและกระชับขึ้นสำหรับใช้งานกับอุปกรณ์พกพาอย่าง โน้ตบุ๊ก ดังภาพที่ 1.2

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 1.2 คีย์บอร์ดติดตั้งภายในของโน้ตบุ๊ก
ที่มา ดัดแปลงจาก (PortableApps. 2012)

คีย์บอร์ดเออร์โกโนมิกส์ (Ergonomic keyboard) เป็นคีย์บอร์ดที่ถูกออกแบบมาให้
เหมาะกับสรีระของมนุษย์ เหมาะกับการใช้งานนาน ๆ ซึ่งจะช่วยลดอาการบาดเจ็บของข้อมือได้
เป็นอย่างดี ดังภาพที่ 1.3



ภาพที่ 1.3 คีย์บอร์ดเออร์โกโนมิกส์
ที่มา (Truly Ergonomic. n.d.)

คีย์บอร์ดไร้สาย (Cordless keyboard) เป็นเทคโนโลยีที่ช่วยลดความยุ่งยากของการ
ต่อสาย หรือตำแหน่งที่ไกลเกินไปของตัวคอมพิวเตอร์กับโต๊ะวางคีย์บอร์ด โดยเทคโนโลยีนี้มี 2 ทาง คือ
ใช้เทคโนโลยีอินฟราเรด (Infrared) และ เทคโนโลยีคลื่นความถี่วิทยุ (Radio frequency) ซึ่งทั้งคู่ใช้
พลังงานจากแบตเตอรี่ในการส่งและรับสัญญาณ ดังภาพที่ 1.4



ภาพที่ 1.4 คีย์บอร์ดไร้สาย
ที่มา (Amazon. n.d.)

คีย์บอร์ดพกพา (Portable keyboard) เป็นคีย์บอร์ดขนาดเล็กหรือพับเก็บได้สำหรับเคลื่อนย้ายไปทำงานนอกสถานที่ ซึ่งอาจเห็นใช้กับโทรศัพท์บางรุ่น ดังภาพที่ 1.5



ภาพที่ 1.5 คีย์บอร์ดพกพา
ที่มา (Amazon. n.d.)

1.1.2 ประเภทชี้ตำแหน่งและควบคุมทิศทาง (Pointing devices) เป็นอุปกรณ์ที่ช่วยสำหรับการชี้ตำแหน่งและควบคุมทิศทางการทำงานของข้อมูล รวมถึงสั่งการให้คอมพิวเตอร์ทำงาน ซึ่งบางคำสั่งอาจมีการโต้ตอบกันระหว่างผู้ใช้กับคอมพิวเตอร์ เช่น ตัด, คัดลอก เป็นต้น

เมาส์แบบทั่วไป (Mechanical mouse) มีการออกแบบโดยการใช้ลูกบอลเป็นตัวขับเคลื่อนทิศทาง มีลักษณะเป็นลูกกลม ๆ ทำจากแผ่นยางกึ่งอยู่ส่วนด้านล่าง ส่วนบนของเมาส์มีปุ่มให้เลื่อกกดประมาณ 2 - 3 ปุ่ม ขึ้นอยู่กับบริษัทผู้ผลิต และมีการพัฒนาให้สามารถควบคุมการทำงานขึ้นลงของสโครลบาร์ในหน้าต่างของโปรแกรมบางประเภทได้เรียกว่าสโครลเมาส์ (Scroll mouse) แต่มีข้อเสียที่เมื่อใช้ไปนาน ๆ จะมีสิ่งสกปรกเข้าไปทำให้การเคลื่อนที่ของลูกบอลที่เป็นตัวขับเคลื่อนทิศทางทำงานได้ไม่ค่อยดี ทำให้ปัจจุบันไม่เป็นที่นิยมใช้ ดังภาพที่ 1.6



ภาพที่ 1.6 เมาส์แบบทั่วไป
ที่มา (Dell Inc. n.d.).

เมาส์แบบแสง หรือออปติคัลเมาส์ (Optical mouse) เมาส์ชนิดนี้ใช้แสงส่องกระทบพื้นผิวและแปลงทิศทางชี้ตำแหน่งแทนการใช้ล้อหมุน ซึ่งช่วยลดปัญหาเรื่องสิ่งสกปรกหรือฝุ่นละอองจับอันเป็นสาเหตุให้การบังคับทิศทางผิดพลาด ซึ่งปัจจุบันมีทั้งแบบใช้สายและไม่ใช้สาย ดังภาพที่ 1.7



ภาพที่ 1.7 เมาส์แบบแสง
ที่มา (Alibaba. n.d.)

แทรคบอล (Trackball) มีหลักการการทำงานคล้ายกับเมาส์ทั่วไป โดยมีลูกบอลติดตั้งไว้
อยู่ส่วนบนเพื่อใช้ในการควบคุมทิศทาง ดังภาพที่ 1.8



ภาพที่ 1.8 แทรคบอล
ที่มา (Amazon. n.d.)

แผ่นรองสัมผัส หรือทัชแพด (Touch pad) เป็นอุปกรณ์ที่ติดอยู่กับคอมพิวเตอร์แบบ
พกพา ส่วนมากจะติดตั้งไว้บริเวณด้านล่างของแป้นพิมพ์ในคอมพิวเตอร์แบบพกพา มีลักษณะเป็นแผ่น
สี่เหลี่ยมบาง ๆ ใช้ทำงานแทนเมาส์โดยการกดสัมผัสหรือใช้นิ้วลากผ่าน ดังภาพที่ 1.9



ภาพที่ 1.9 แผ่นรองสัมผัส
ที่มา (Redmond Pie. n.d.)

จอยสติ๊ก (Joystick) และพวงมาลัยบังคับทิศทาง (Wheel) พบเห็นได้กับการใช้เล่น
เกมคอมพิวเตอร์เพื่อให้เกิดความเสมือนจริงในการควบคุมบังคับทิศทาง ดังภาพที่ 1.10



ภาพที่ 1.10 จอยสติ๊ก
ที่มา (Pelaajan. 2009)

จอสัมผัสหรือทัชสกรีน (Touch screen) เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อใช้ทำงานแทนเมาส์และคีย์บอร์ดตลอดจนยังสร้างปฏิสัมพันธ์กับผู้ใช้ (Interactive) ได้เป็นอย่างดี โดยสามารถใช้นิ้วมือแตะบังคับลงไปยังหน้าจอกอมพิวเตอร์ได้เลย ดังภาพที่ 1.11



ภาพที่ 1.11 จอสัมผัสหรือทัชสกรีน
ที่มา (SlashGea. n.d.)

1.1.3 ประเภทปากกา (Pen - based device)

ปากกาแสง (Light pen) เป็นอุปกรณ์สำหรับเขียนหรือวาดมักใช้ร่วมกับโปรแกรมประเภทช่วยสำหรับการออกแบบ (CAD: Computer aided design) ตลอดจนเป็นการกำหนดตำแหน่งบนจอภาพรวมถึงการป้อนข้อมูลเข้าแทนแป้นพิมพ์ แต่มีความละเอียดไม่มากทำให้ปัจจุบันไม่เป็นที่นิยม ดังภาพที่ 1.12



ภาพที่ 1.12 ปากกาแสง
ที่มา (Peter & Kim. 1979)

สไตลัส (Stylus) เป็นปากกาที่นิยมใช้กันมากในคอมพิวเตอร์ขนาดเล็ก เช่น พีดีเอ แท็บเล็ตพีซี หรือสมาร์ทโฟน ซึ่งสามารถใช้เลือกรายการ เขียนตัวหนังสือหรือวาดลายเส้นลงบนหน้าจออุปกรณ์นั้น ๆ ได้โดยตรง ดังภาพที่ 1.13



ภาพที่ 1.13 สไตลัส
ที่มา (iPhone JD. 2011)

ดิจิไทเซอร์ (Digitizer) เป็นปากกาที่ใช้ในงานที่มีความละเอียดสูง มีความคล่องตัวและสะดวกต่อการใช้งานมาก ใช้ร่วมกับกระดานรองรับสำหรับการป้อนข้อความ วาดภาพหรือออกแบบงานที่เกี่ยวข้องกับกราฟิก ดังภาพที่ 1.14



ภาพที่ 1.14 ดิจิไทเซอร์
ที่มา (Weebly. n.d.)

1.1.4 ประเภทข้อมูลมัลติมีเดีย (Multimedia input device)

ไมโครโฟน (Microphone) เป็นอุปกรณ์ที่ช่วยรับข้อมูลประเภทเสียงเข้าสู่คอมพิวเตอร์ ซึ่งจะใช้ร่วมกับซอฟต์แวร์ด้านมัลติมีเดีย ซึ่งปัจจุบันกูเกิ้ล (Google) ได้พัฒนาให้สามารถใช้เสียงพูดแทนระบบการพิมพ์ได้ ดังภาพที่ 1.15



ภาพที่ 1.15 ไมโครโฟน
ที่มา (AltoEdge. n.d.)

กล้องถ่ายภาพดิจิทัล (Digital camera) เป็นอุปกรณ์ที่ช่วยรับข้อมูลประเภทภาพถ่ายดิจิทัล เพื่อนำมาปรับแต่งภาพ หรือปรับปรุงภาพเพื่อให้ได้ภาพที่สวยงามหรือตรงกับความต้องการของผู้ใช้ ทั้งนี้ความละเอียดของภาพจะมีความแตกต่างกันไปตามชนิดของกล้อง ดังภาพที่ 1.16



ภาพที่ 1.16 กล้องถ่ายภาพดิจิทัล
ที่มา (CircuitsToday. n.d.)

กล้องถ่ายวิดีโอดิจิทัล (Digital video camera) เป็นอุปกรณ์ที่ช่วยรับข้อมูลประเภทภาพเคลื่อนไหวเพื่อนำมาปรับแต่ง ตัดต่อให้ได้เรื่องราวตามความต้องการของผู้ใช้ ทั้งนี้ความละเอียดของภาพจะมีความแตกต่างกันไปตามชนิดของกล้อง ดังภาพที่ 1.17



ภาพที่ 1.17 กล้องถ่ายวิดีโอดิจิทัล
ที่มา (Media data systems pte ltd. n.d.)

เว็บแคม (Web cam) เป็นอุปกรณ์ที่ช่วยรับข้อมูลประเภทภาพเคลื่อนไหวเช่นเดียวกับกล้องถ่ายวิดีโอดิจิทัล แต่คุณภาพที่ได้หยาบและมีขนาดไฟล์เล็กกว่า ทั้งนี้ด้วยไฟล์ที่มีขนาดเล็กจึงส่งผลให้สามารถติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตได้ดี แลมนี้อาจถูกกว่ากล้องถ่ายวิดีโอดิจิทัล ดังภาพที่ 1.18



ภาพที่ 1.18 เว็บแคม
ที่มา (Ziff davis. n.d.)

1.1.5 ประเภทสแกนและอ่านข้อมูลด้วยแสง
สแกนเนอร์ (Scanner) เป็นอุปกรณ์สำหรับอ่านข้อมูลประเภทภาพถ่ายหรือเอกสารที่เป็นชิ้นงาน ทำงานคล้ายเครื่องถ่ายภาพเอกสารแต่อ่านภาพเป็นไฟล์เพื่อใช้สำหรับส่งข้อมูลเข้าคอมพิวเตอร์ต่อไป ดังภาพที่ 1.19



ภาพที่ 1.19 สแกนเนอร์
ที่มา (Google sites. n.d.)

โอเอ็มอาร์ (OMR: Optical mark reader) เป็นอุปกรณ์ที่ใช้ในการตรวจสอบหรือคะแนนของกลุ่มบุคคลจำนวนมาก โดยเครื่องจะอ่านเครื่องหมายที่ผู้เข้าสอบได้ระบายไว้ในกระดาษคำตอบ เช่น การสอบเอ็นทรานซ์ การสอบวัดระดับภาษาอังกฤษ ดังภาพที่ 1.20



ภาพที่ 1.20 โอเอ็มอาร์
ที่มา (Chatsworth. 2010)

เครื่องอ่านบาร์โค้ด (Bar code reader) เป็นอุปกรณ์ที่ พบเห็นได้ตามจุดบริการเก็บเงิน (POS: Point of sale) ร้านอาหาร ร้านสะดวกซื้อ หรือห้างสรรพสินค้าทั่วไป ใช้แทนการป้อนค่าสินค้าที่มีตัวเลขวัสดุติดอยู่ เพื่อลดการผิดพลาดในการพิมพ์ ดังภาพที่ 1.21



ภาพที่ 1.21 เครื่องอ่านบาร์โค้ด
ที่มา (Barcode118. 2014)

เอ็มไอซีอาร์ (MICR: Magnetic ink character recognition) เป็นอุปกรณ์ที่ใช้แสงอ่านตัวอักษรของเอกสารสำคัญพบเห็นได้กับการประมวลผลเช็คในธุรกิจด้านธนาคารเป็นส่วนใหญ่ ดังภาพที่ 1.22



ภาพที่ 1.22 เอ็มไอซีอาร์
ที่มา (Encompos. n.d.)

1.1.6 ประเภทตรวจสอบข้อมูลทางกายภาพ

ไบโอเมตริกส์ (Biometric) เป็นอุปกรณ์ช่วยป้องกันและรักษาความปลอดภัยในระดับที่สูงมากโดยการตรวจสอบข้อมูลเฉพาะตัวของแต่ละบุคคล เช่น เครื่องอ่านลายนิ้วมือ เครื่องตรวจสอบม่านตา เครื่องวิเคราะห์เสียงพูด ดังภาพที่ 1.23



ภาพที่ 1.23 ไบโอมेटริกส์
ที่มา (Bio office. n.d.)

1.2 อุปกรณ์ฮาร์ดแวร์ของหน่วยประมวลผล เป็นอุปกรณ์ที่เกี่ยวข้องกับการประมวลผล ข้อมูลที่รับเข้ามา ก่อนที่จะจัดเก็บหรือแสดงผล ซึ่งแบ่งเป็นประเภทตามที่พบเห็นในปัจจุบันได้ดังนี้

1.2.1 ซีพียู (CPU: Central processing unit) เป็นอุปกรณ์ภายในคอมพิวเตอร์ที่คอยวิเคราะห์ ยิ่งซีพียูเร็วมากเท่าไร การประมวลผลก็ยิ่งเร็วขึ้นเท่านั้น โดยมีสัญญาณคล็อก (Clock) เป็นตัวบอกจังหวะการทำงาน ยิ่งคล็อก มีค่ามากก็จะทำให้คอมพิวเตอร์เครื่องนั้นมีความเร็วสูงตามไปด้วย ซึ่งมีหน่วยวัดเป็น “เฮิร์ตซ์” (Hertz) เช่น ซีพียูคอร์ไอเซเวน (Core i7) มีความเร็ว 2.8 จิกกะเฮิร์ตซ์หมายถึงการทำงานที่ความเร็ว 2,800 ล้านครั้ง ในหนึ่งวินาที ดังภาพที่ 1.24

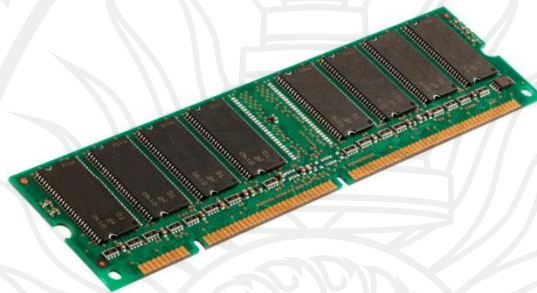


ภาพที่ 1.24 ซีพียู
ที่มา (จักรกรฤษณ์. ม.ป.ป.)

1.2.2 แรม โดยทั่วไปสามารถแบ่งได้เป็น 2 ประเภทใหญ่คือ (ยงยุทธ, ม.ป.ป.)
แรมสถิต (SRAM: Static RAM) มีความเร็วในการทำงานสูงกว่า แรมพลวัต (DRAM: Dynamic RAM) จึงนิยมนำไปใช้เป็นหน่วยแคช (Cache) ภายในตัวซีพียู แต่ไม่สามารถทำให้มีขนาดความจุสูง ๆ ได้เพราะการกินกระแสไฟที่มากจนทำให้เกิดความร้อนสูง

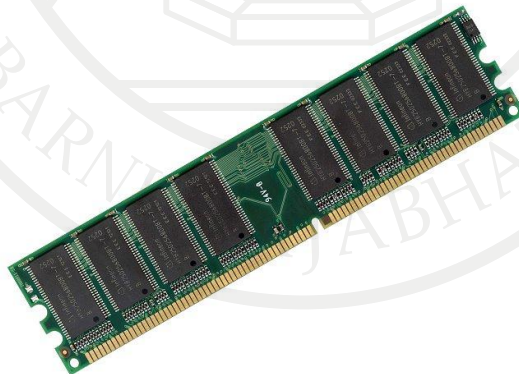
แรมพลวัต (DRAM: Dynamic RAM) ได้ถูกออกแบบให้มีขนาดความจุสูง ๆ ได้ กินไฟน้อยและไม่เกิดความร้อนสูง และมีหลากหลายชนิด โดยชนิดของแรมพลวัตหรือดีแรม (DRAM) ที่นำมาใช้ทำเป็นแผงหน่วยความจำหลักของระบบชนิดต่าง ๆ ในปัจจุบันดังนี้

1) เอสดีแรม (SDRAM: Synchronous DRAM) ใช้บรรจุภัณฑ์แบบทีเอสโอพี (TSOP: Thin small outline package) ติดตั้งอยู่บน แผงโมดูลแบบดีไอเอ็มเอ็ม (DIMM: Dual inline memory module) ที่มีร่องบากบริเวณแนวขาสัญญาณ 2 ร่อง และมีจำนวนขาทั้งสิ้น 168 ขา ใช้แรงดันไฟ 3.3 โวลต์ ความเร็วบัสมีให้เลือกใช้ทั้งรุ่น พีซี - 66 (66 เมกกะเฮิร์ต), พีซี - 100 (100 เมกกะเฮิร์ต), พีซี - 133 (133 เมกกะเฮิร์ต) และ พีซี - 150 (150 เมกกะเฮิร์ต) แต่ปัจจุบันหมดความนิยมไปแล้ว จะพบได้เพียงในคอมพิวเตอร์รุ่นเก่า ๆ เท่านั้น ดังภาพที่ 1.25



ภาพที่ 1.25 เอสดีแรม
ที่มา (เกียรติศักดิ์. 2552)

2) ดิดีอาร์ เอสดีแรม (DDR SDRAM: Double data rate SDRAM) ตัวชิปจะใช้บรรจุภัณฑ์แบบทีเอสโอพีเช่นเดียวกับเอสดีแรมและมีขนาด ความยาวของแผงโมดูลเท่ากัน คือ 5.25 นิ้ว ติดตั้งอยู่บนแผงโมดูลแบบดีไอเอ็มเอ็มที่มีร่องบากบริเวณแนวขาสัญญาณ 1 ร่อง และมีจำนวนขาทั้งสิ้น 184 ขา ใช้แรงดันไฟ 2.5 โวลต์ ความเร็วบัสในปัจจุบันมีให้เลือกใช้ตั้งแต่ 133 เมกกะเฮิร์ตไปจนถึง 350 เมกกะเฮิร์ต สำหรับดีแรมชนิดนี้ปัจจุบันกำลังจะหมดความเป็นที่นิยมเช่นกัน ดังภาพที่ 1.26



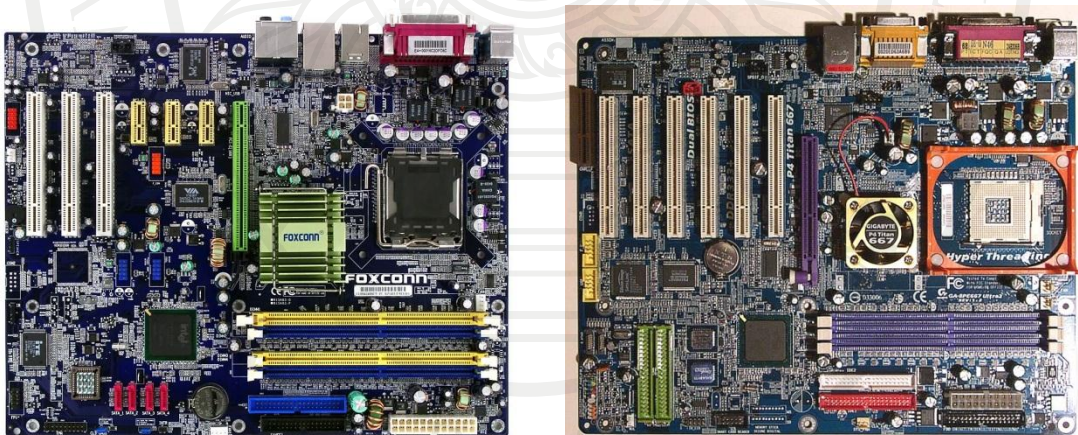
ภาพที่ 1.26 ดิดีอาร์ เอสดีแรม
ที่มา (เกียรติศักดิ์. 2552)

3) ดีดิวอาร์ทู เอสดีแรม (DDR-II SDRAM) ตัวชิปจะใช้บรรจุภัณฑ์แบบเอฟบีจีเอ (FBGA: Fine pitch ball grid array) ที่มีความต้านทานไฟฟ้าต่ำกว่าแบบทีเอสโอพี อีกทั้งยังสามารถออกแบบให้ตัวชิปมีขนาดเล็กและบางลงได้ ชิปดังกล่าวถูกติดตั้งอยู่บนแผงโมดูลแบบดีไอเอ็มเอ็มที่มีร่องบากบริเวณแนวขาสัญญาณ 1 ร่อง และมีจำนวนขาทั้งสิ้น 240 ขา ใช้แรงดันไฟเพียง 1.8 โวลต์ ความเร็วบัสในปัจจุบันมีให้เลือกใช้ตั้งแต่ 200 เมกกะเฮิร์ต ไปจนถึง 450 เมกกะเฮิร์ต สำหรับดีแรมชนิดนี้ปัจจุบันกำลังได้รับความนิยมเป็นอย่างมาก จนคาดว่าในอีกไม่ช้าจะเข้ามาแทนที่มาตรฐานเดิมคือ ดีดิวอาร์ เอสดีแรมในที่สุด ดังภาพที่ 1.27



ภาพที่ 1.27 ดีดิวอาร์-ทู เอสดีแรม
ที่มา (เกียรตีสักดี. 2552)

1.2.3 เมนบอร์ด (Mainboard) เป็นศูนย์กลางของการเชื่อมต่อสำหรับอุปกรณ์ภายในเครื่องคอมพิวเตอร์ บางครั้งถูกเรียกว่าเป็นแผงวงจรหลัก ที่คอยส่งสัญญาณไฟฟ้าให้อุปกรณ์ต่าง ๆ ภายในเครื่องทำงานร่วมกัน ดังภาพที่ 1.28



ภาพที่ 1.28 เมนบอร์ด
ที่มา (ณรงค์. 2547)

1.2.4 ฮาร์ดดิสก์ (Harddisk) เป็นอุปกรณ์ที่คอยเก็บบันทึกข้อมูล ผลิตจากวัสดุแบบแข็งจำนวนหลายแผ่นวางเรียงต่อกันเป็นชั้น ดังภาพที่ 1.29



ภาพที่ 1.29 ฮาร์ดดิสก์
ที่มา (นิรนาม. 2556)

1.3 อุปกรณ์ฮาร์ดแวร์ของหน่วยแสดงผล เป็นอุปกรณ์ที่ได้จากการประมวลผลของคอมพิวเตอร์ที่มีทั้งตัวอักษร ภาพและเสียง

1.3.1 อุปกรณ์แสดงผลหน้าจอสำหรับแสดงผลกราฟฟิกต่าง ๆ

พอยท์ออฟเซลเทอร์มินัล (Point of sale terminal) เป็นจอภาพที่มีขนาดขนาดเล็กกว่าจอภาพของคอมพิวเตอร์ทั่วไป พบเห็นได้ตามจุดบริการขาย หรือเพื่อทำรายการต่าง ๆ อย่างตู้เอทีเอ็ม ดังภาพที่ 1.30



ภาพที่ 1.30 พอยท์ออฟเซลเทอร์มินัล
ที่มา (Pallab chatterjee. 2012)

จอซีอาร์ที (CRT: Cathode ray tube) เป็นจอรุ่นเก่า มีลักษณะจอภาพเป็นกล่องสี่เหลี่ยม การทำงานเป็นเหมือนจอโทรทัศน์ ใช้การฉายแสงอิเล็กตรอนของหลอดภาพในการแสดงผล โดยแสงที่เกิดขึ้นเกิดจากการผสมสีสามสีได้แก่ สีแดง เขียว และ น้ำเงิน ซึ่งในการยิงแสงแต่ละครั้ง อาจจะดูเหมือนสั้นตลอดเวลา ภาพอาจไม่นิ่งและทำให้ปวดตาในที่สุด และเนื่องจากการทำงานดังกล่าว ทำให้เกิดความร้อน การใช้พลังงานที่สูงมาก มีขนาดใหญ่ น้ำหนักมาก และยังมีรังสีแผ่กระจายออกมา ปัจจุบันยกเลิกการผลิตไปแล้ว จะเหลือขายเพียงแค่มือสองเท่านั้น ซึ่งเมื่อก่อนจอลักษณะนี้นิยมใช้กับเครื่องคอมพิวเตอร์ประเภทพีซี ดังภาพที่ 1.31



ภาพที่ 1.31 จอซีอาร์ที
ที่มา (นิรนาม. 2555)

จอแอลซีดี (LCD: Liquid crystal display) เป็นจอแบนสี่เหลี่ยมอาศัยการทำงานของโมเลกุลที่เรียกว่าผลึกเหลวในการแสดงผล แต่เนื่องจากมีข้อจำกัดอย่างเช่น การมองไม่ถูกมุมอาจทำให้สีที่เราเห็นนั้นผิดไปและไม่ชัดในบางมุม ตลอดจนอาจแสดงผลที่ช้ากว่าจอซีอาร์ที จึงมีการระบุความเร็วในการแสดงผลไว้กับรายละเอียดการเลือกซื้อ ระยะเวลาในการใช้งานมีข้อจำกัด แต่ด้วยไม่มีรังสีใช้ ความร้อนและพลังงานน้อยกว่า จึงนิยมใช้กับเครื่องคอมพิวเตอร์ประเภทโน้ตบุ๊ก และปัจจุบันเริ่มนิยมใช้กับเครื่องคอมพิวเตอร์ประเภทพีซีมากขึ้น ดังภาพที่ 1.32



ภาพที่ 1.32 จอแอลซีดี
ที่มา (นิรนาม. 2555)

โปรเจคเตอร์ (Projector) นิยมใช้สำหรับการจัดประชุม สัมมนา การเรียนการสอน หรือการนำเสนอผลงานต่าง ๆ ให้ผู้ชมจำนวนมากได้เห็น ดังภาพที่ 1.33



ภาพที่ 1.33 โปรเจคเตอร์
ที่มา (SV Plannet Ltd. n.d.)

1.3.2 อุปกรณ์สำหรับพิมพ์งาน

เครื่องพิมพ์แบบดอทเมทริกซ์ หัวพิมพ์มีลักษณะเป็นหัวเข็มทำงานโดยการสร้างจุดลงบนกระดาษ ด้วยการกระแทกกับผ้าหมึกลงกระดาษที่ใช้พิมพ์ เครื่องพิมพ์ประเภทนี้ยังเป็นที่นิยมกันสำหรับทำใบเสร็จ แต่ข้อเสียคือเวลาสั่งพิมพ์จะเกิดเสียงดังพอสมควร ดังภาพที่ 1.34



ภาพที่ 1.34 เครื่องพิมพ์แบบดอทเมทริกซ์
ที่มา (Officemate. n.d.)

เครื่องพิมพ์แบบเลเซอร์ เป็นเครื่องพิมพ์ที่ใช้เทคโนโลยีเดียวกับเครื่องถ่ายเอกสาร คือยิงเลเซอร์ให้ความร้อนไปบนกระดาษและให้ผงหมึกติดกระดาษขึ้นมา ทำให้งานที่ได้มีคุณภาพดีมาก และทำงานได้เร็วกว่าเครื่องพิมพ์แบบพ่นหมึก แต่ก็มีราคาที่สูงมากเช่นกัน ดังภาพที่ 1.35



ภาพที่ 1.35 เครื่องพิมพ์แบบเลเซอร์
ที่มา (Driverbasket. n.d.)

เครื่องพิมพ์แบบพ่นหมึก หรือเครื่องพิมพ์แบบอิงค์เจ็ต (Inkjet printer) เมื่อต้องการพิมพ์รูปทรงหรือรูปภาพใด ๆ เครื่องพิมพ์จะทำงานโดยการพ่นหมึกออกมาเป็นหยดเล็ก ๆ ลงบนกระดาษ ตามความต้องการของเรา ซึ่งเครื่องพิมพ์แบบพ่นหมึกจะมีคุณภาพดีกว่าเครื่องพิมพ์ดอตแมทริกซ์ และมีราคาถูกกว่าเครื่องพิมพ์แบบเลเซอร์ ดังภาพที่ 1.36



ภาพที่ 1.36 เครื่องพิมพ์แบบพ่นหมึก
ที่มา (Notebookspec. 2011)

1.3.3 อุปกรณ์ขับเสียง

ลำโพง การแสดงผลนอกเหนือจากตัวอักษรและภาพนั้นคือเสียง ลำโพงจึงเป็นอุปกรณ์หนึ่งที่จะช่วยในการแสดงผล ดังภาพที่ 1.37



ภาพที่ 1.37 ลำโพง
ที่มา (Sajee. 2013)

หูฟัง เป็นอุปกรณ์เช่นเดียวกับลำโพงต่างกันตรงที่สวมใส่ไว้ข้างหู นิยมใช้สำหรับการฟังคนเดียว ปัจจุบันมีแบบไร้สายและราคาถูกลงมาก ดังภาพที่ 1.38



ภาพที่ 1.38 หูฟัง
ที่มา (Hearbeats. n.d.)

2. ซอฟต์แวร์ (Software) จัดเป็นส่วนประกอบสำคัญที่ขาดไม่ได้สำหรับการทำงานกับคอมพิวเตอร์ เพราะซอฟต์แวร์ทำหน้าที่สั่งให้คอมพิวเตอร์ ประมวลผลหรือเปรียบเทียบค่าต่าง ๆ ตามที่ผู้พัฒนาซอฟต์แวร์ต้องการ เช่น ช่วยคำนวณแบบหรือโครงสร้างที่ซับซ้อนในงานวิศวกรรม การจำลองและแสดงผลการเคลื่อนที่ของเครื่องจักรต่าง ๆ หรือแม้แต่การจัดการฐานข้อมูลสำหรับงานสารสนเทศ เป็นต้น โดยชนิดของซอฟต์แวร์แบ่งได้เป็น 2 ส่วนใหญ่ ๆ คือ (วิโรจน์และสุพรรณษา, 2552)

2.1 ซอฟต์แวร์ระบบ เป็นซอฟต์แวร์ที่ถูกสร้างขึ้นเพื่อจัดการระบบ โดยจัดการเกี่ยวกับการดำเนินงานพื้นฐานต่าง ๆ ของคอมพิวเตอร์ เช่น รับข้อมูลจากเมาส์หรือคีย์บอร์ดแล้วแปลความหมายให้คอมพิวเตอร์ ตลอดจนนำข้อมูลออกไปแสดงผลยังเครื่องพิมพ์หรือจอภาพ เป็นต้น อีกทั้งซอฟต์แวร์ระบบยังช่วยจัดการข้อมูลบนหน่วยความจำสำรอง ระบบแฟ้มข้อมูล การเก็บข้อมูลไว้ในรอมหรือในแผ่นจานแม่เหล็ก ทั้งนี้คอมพิวเตอร์อาจทำงานไม่ได้หากไม่มีซอฟต์แวร์ระบบ ซึ่งซอฟต์แวร์ระบบที่เห็นกันทั่วไป ยังสามารถแบ่งออกได้อีก 2 ส่วนคือระบบปฏิบัติการ และตัวแปลภาษา

2.1.1 ระบบปฏิบัติการ (OS: Operating system) หรือ โอเอส เป็นซอฟต์แวร์ระบบที่ใช้ดูแลระบบคอมพิวเตอร์ และจำเป็นมากสำหรับเครื่องคอมพิวเตอร์ทุกเครื่อง โดยระบบปฏิบัติการที่นิยมใช้กันมากและเป็นที่รู้จักกันดีเช่น

2.1.1.1 ดอส (DOS: Disk operating system) เป็นซอฟต์แวร์จัดระบบงานที่ปัจจุบันมีการใช้งานน้อยมากเนื่องจากการพัฒนาของซอฟต์แวร์ระบบ โดยดอสเป็นซอฟต์แวร์ที่รู้จักกันดีในหมู่ผู้ใช้ไมโครคอมพิวเตอร์

2.1.1.2 วินโดวส์ (Windows) เป็นระบบปฏิบัติการที่พัฒนาต่อดอส มีการเชื่อมต่อกับเมาส์ ทำให้ผู้ใช้สามารถใช้งานได้หลากหลายและง่ายขึ้น โดยงานจะอยู่ในกรอบช่องหน้าต่างที่แสดงผลบนจอภาพ เน้นให้สามารถติดต่อกับผู้ใช้ได้ง่ายขึ้นโดยมีรูปแบบกราฟิกเข้ามาช่วยทำให้วินโดวส์ได้รับความนิยมในปัจจุบัน

2.1.1.3 ยูนิกซ์ (Unix) เป็นระบบปฏิบัติการที่เป็นเทคโนโลยีแบบเปิด (Open system) ถูกพัฒนามาตั้งแต่ใช้กับเครื่องมินิคอมพิวเตอร์ โดยผู้ใช้ไม่ต้องผูกติดกับระบบใดระบบหนึ่ง

หรือใช้อุปกรณ์ที่มีฮาร์ดแวร์เหมือนกัน อีกทั้งยูนิคซ์ยังถูกออกแบบมาสำหรับการใช้งานได้หลายคนในเวลาเดียวกันหรือระบบหลายผู้ใช้ (Multiusers) ตลอดจนทำงานได้หลาย ๆ งานในเวลาเดียวกันหรือระบบหลายภารกิจ (Multitasking) ดังนั้นยูนิคซ์จึงเป็นระบบปฏิบัติการที่นิยมใช้กับเครื่องที่เชื่อมโยงเป็นเครือข่าย ที่ใช้งานร่วมกันหลาย ๆ เครื่องพร้อมกัน

2.1.1.4 ลินุกซ์ (Linux) เป็นระบบปฏิบัติการที่พัฒนาจากระบบยูนิคซ์ ซึ่งระบบลินุกซ์เป็นระบบปฏิบัติการประเภทแจกฟรี (Free ware) ผู้ใช้สามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย โดยระบบนี้มีการแจกจ่ายโปรแกรมต้นฉบับให้นักพัฒนาช่วยกันพัฒนาคุณสมบัติของระบบปฏิบัติการ มีโปรแกรมประยุกต์ต่าง ๆ ที่ทำงานบนระบบลินุกซ์จำนวนมาก และระบบปฏิบัติการลินุกซ์เป็นที่นิยมกันมากขึ้นในปัจจุบัน โดยเฉพาะอย่างยิ่งโปรแกรมในกลุ่มของกนู (GNU)

2.1.1.5 แมคอินทอช (Macintosh) เป็นระบบปฏิบัติการสำหรับเครื่องไมโครคอมพิวเตอร์แมคอินทอช นิยมใช้ในงานด้านออกแบบ กราฟิก ตลอดจนใช้ในสำนักพิมพ์ต่าง ๆ

นอกจากระบบปฏิบัติการที่กล่าวมาแล้วยังมีระบบปฏิบัติการอื่น ๆ อีกมาก เช่น ระบบปฏิบัติการที่ใช้ในเครือข่ายคอมพิวเตอร์เพื่อให้คอมพิวเตอร์ทำงานร่วมกันเป็นระบบอย่างระบบปฏิบัติการเน็ตแวร์ เป็นต้น

2.1.2 ตัวแปลภาษา การติดต่อกับคอมพิวเตอร์จำเป็นต้องมีการแปลภาษาระดับสูงให้เป็นภาษาเครื่อง ทั้งนี้ภาษาระดับสูงมีหลายภาษาซึ่งถูกสร้างขึ้นเพื่อให้ผู้เขียนโปรแกรมเขียนชุดคำสั่งหรือปรับปรุงแก้ไขซอฟต์แวร์ได้ง่ายในภายหลัง ซึ่งภาษาระดับสูงที่พัฒนาขึ้นจนเป็นที่รู้จักและนิยมกันมากในปัจจุบัน เช่น ภาษาซี ภาษาเบสิก และภาษาปาสคาล

2.1.2.1 ภาษาซี (C programming language) คือ ภาษาคอมพิวเตอร์ใช้สำหรับพัฒนาโปรแกรมทั่วไป ถูกพัฒนาครั้งแรกเพื่อใช้เป็นภาษาสำหรับพัฒนาระบบปฏิบัติการยูนิคซ์ (Unix operating system) แทนภาษาแอสเซมบลี

2.1.2.2 ภาษาเบสิก (Beginner's allpurpose symbolic instruction code) เป็นภาษาที่สามารถเรียนรู้และเข้าใจได้ง่าย มีรูปแบบคำสั่งพื้นฐานที่สามารถนำมาเขียนเรียงต่อกันเป็นโปรแกรมได้ มีรูปแบบคำสั่งไม่ยุ่งยาก โดยบริษัทไมโครซอฟท์ได้นำภาษาเบสิกมาปรับปรุงให้ทันสมัย และพัฒนาเครื่องมือในโปรแกรมวิซวลเบสิก ทำให้ภาษาเบสิกได้รับความนิยมในการพัฒนาโปรแกรมยุคใหม่

2.1.2.3 ภาษาปาสคาล เป็นภาษาโปรแกรมที่ใช้กันอย่างกว้างขวาง โดยเฉพาะในวงการศึกษ เป็นภาษาสั่งงานคอมพิวเตอร์ที่มีรูปแบบเป็นโครงสร้าง ผู้เขียนสามารถแบ่งแยกงานออกเป็นขั้นเล็ก ๆ แล้วมารวมกันเป็นโปรแกรมขนาดใหญ่ได้

นอกจากนี้ยังมีภาษาคอมพิวเตอร์ที่ใช้กันอยู่ในปัจจุบันอีกมากมายหลายภาษา เช่น ภาษาฟอร์แทรน ภาษาโคบอล ภาษาอาร์พีจี เป็นต้น

2.2 ซอฟต์แวร์ประยุกต์ ในปัจจุบันมีผู้พัฒนาซอฟต์แวร์ในงานด้านต่าง ๆ มากมาย ที่ใช้กับงานตามความต้องการของผู้ใช้ เพื่อก่อให้เกิดประโยชน์หรือทำให้ทำงานได้สะดวกขึ้น โดยซอฟต์แวร์ประยุกต์สามารถแบ่งออกเป็นสองกลุ่มคือ ซอฟต์แวร์สำเร็จ และซอฟต์แวร์ที่พัฒนาขึ้นใช้งานเฉพาะ ซอฟต์แวร์สำเร็จในปัจจุบันมีมากมาย เช่น ซอฟต์แวร์ประมวลคำ ซอฟต์แวร์ตารางทำงาน ฯลฯ

2.2.1 ซอฟต์แวร์สำเร็จ (Package) เป็นซอฟต์แวร์ที่บริษัทพัฒนาขึ้นแล้วนำออกมาจำหน่าย เพื่อให้ผู้ใช้งานซื้อไปใช้ได้โดยตรง โดยไม่ต้องเสียเวลาในการพัฒนาซอฟต์แวร์อีก ซอฟต์แวร์

สำเร็จที่มีจำหน่ายในท้องตลาดทั่วไป และเป็นที่ยอมรับของผู้ใช้ส่วนใหญ่ลงไว้ประจำเครื่องได้แก่ ซอฟต์แวร์ประมวลคำ (Word processing software) ซอฟต์แวร์ตารางคำนวณ (Spread sheet software) ซอฟต์แวร์จัดการฐานข้อมูล (Data base management software) ซอฟต์แวร์นำเสนอ (Presentation software)

2.2.1.1 ซอฟต์แวร์ประมวลคำ เป็นซอฟต์แวร์ที่ใช้สำหรับการพิมพ์เอกสาร สามารถจัดการเอกสารต่าง ๆ ได้ เช่น แก้ไขขนาดตัวอักษร เพิ่มข้อความ แทรกรูปภาพ เป็นต้น ทำเอกสารให้ดูเรียบร้อยสวยงาม สามารถแสดงผลการพิมพ์ออกทางเครื่องพิมพ์ได้ ซึ่งซอฟต์แวร์ประมวลคำที่นิยมอยู่ในปัจจุบัน เช่น ไมโครซอฟเวิร์ด จูพา ปลาตาว ฯลฯ

2.2.1.2 ซอฟต์แวร์ตารางคำนวณ เป็นซอฟต์แวร์ที่ช่วยในการคิดคำนวณ มักนำไปใช้งานเกี่ยวกับตัวเลข เช่น บัญชี โดยตัวซอฟต์แวร์แบ่งเป็นช่องตารางสามารถใส่สูตร ข้อความ ตัวเลขต่าง ๆ เพื่อประมวลผลหรือเปรียบเทียบได้ เช่น ไมโครซอฟเอ็กเซล ฯลฯ

2.2.1.3 ซอฟต์แวร์จัดการฐานข้อมูล เป็นซอฟต์แวร์ที่เกี่ยวข้องกับข้อมูลจำนวนมาก ๆ และการจัดการกับข้อมูลที่จัดเก็บในคอมพิวเตอร์ เช่นช่วยในเรียกใช้ข้อมูล การจัดเรียงข้อมูล การทำรายงาน การสรุปผลจากข้อมูล ซอฟต์แวร์จัดการฐานข้อมูลที่นิยมใช้ เช่น ไมโครซอฟเอ็กเซส ดีเบส ฯลฯ

2.2.1.4 ซอฟต์แวร์นำเสนอ เป็นซอฟต์แวร์ที่ใช้สำหรับช่วยนำเสนอข้อมูลเป็นหลัก ช่วยสร้างสิ่งสวยงามและดึงดูดความสนใจในระหว่างการนำเสนอ เช่น ไมโครซอฟเพาเวอร์พอยต์ ฮาร์วาร์ดกราฟิก ฯลฯ

2.2.2 ซอฟต์แวร์ใช้งานเฉพาะ คือซอฟต์แวร์ที่หน่วยงานหรือองค์กรไม่สามารถจัดหาซอฟต์แวร์ที่นำมาใช้ได้อย่างเหมาะสมหรือมีประสิทธิภาพดีพอกับความต้องการ ซึ่งจากที่กล่าวมาการประยุกต์ใช้งานด้วยซอฟต์แวร์สำเร็จมักจะเน้นการใช้งานทั่วไป แต่อาจจะไม่สามารถนำมาใช้โดยตรงกับธุรกิจบางอย่างได้ เช่น การขายสินค้า กิจกรรมธนาคารที่มีการฝากถอนเงิน งานทางด้านบัญชี หรืองานในห้างสรรพสินค้าที่มีการออกใบเสร็จรับเงิน ฯลฯ ดังนั้นจึงต้องมีการพัฒนาซอฟต์แวร์ให้ใช้งานเฉพาะสำหรับงานแต่ละประเภทให้ตรงกับความต้องการของผู้ใช้แต่ละราย

โดยซอฟต์แวร์ใช้งานเฉพาะนี้ ผู้พัฒนาอาจต้องเข้าไปศึกษารูปแบบการทำงานหรือความต้องการของธุรกิจนั้น ๆ แล้วจัดทำขึ้น เช่น ระบบงานทางด้านบัญชี ระบบงานจัดจำหน่าย ระบบงานในโรงงานอุตสาหกรรม บริหารการเงิน และการเช่าซื้อ ความต้องการของการใช้คอมพิวเตอร์ในงานทางธุรกิจยังมีอีกมาก ดังนั้นจึงมีความต้องการที่จะให้ผู้พัฒนาซอฟต์แวร์ พัฒนาซอฟต์แวร์สำหรับใช้งานเฉพาะ อีกมากมาย

โปรแกรมคอมพิวเตอร์สำหรับงานวิศวกรรมมีด้วยกันหลากหลายชนิด อาทิ โซลิดเวิร์คส์ (Solidworks), ออโต้แคท (AutoCAD), แมทแล็บ(Matlab) ฯลฯ แต่ละโปรแกรมก็ช่วยในงานที่แตกต่างกันไป สำหรับในงานระบบที่ต้องการให้ผู้ใช้งานได้ง่ายนั้นยังคงไม่มีโปรแกรมสำเร็จที่เหมาะสมและใช้ได้ทุกโรงงาน การพัฒนาโปรแกรมหรือออกแบบระบบจึงยังคงเป็นที่ต้องการอยู่ยิ่งกว่า ซึ่งโปรแกรมวิศวกรรมเชิงจัดเป็นอีกหนึ่งโปรแกรมที่มีความสมบูรณ์ในการช่วยพัฒนางานทางด้านวิศวกรรมให้เกิดขึ้น เพราะตัวโปรแกรมประกอบด้วยเครื่องมือที่ช่วยให้การพัฒนาแอปพลิเคชันที่หลากหลาย อีกทั้งสะดวกสบายต่อการใช้งาน จึงได้นำโปรแกรมวิศวกรรมเชิงจัดมาใช้เป็นสื่อในการเรียนการสอนเพื่อให้เกิดความรู้ความเข้าใจในการพัฒนาโปรแกรมสำหรับวิศวกรรมต่อไป

โปรแกรมวิชาการ

โปรแกรมภาษาวิชาการเบสิกเป็นภาษาคอมพิวเตอร์ ที่พัฒนาโดยบริษัทไมโครซอฟท์ (Microsoft) ที่สร้างระบบปฏิบัติการวินโดวส์ที่ใช้กันอยู่ในปัจจุบัน ตัวภาษาดั้งเดิมมาจากภาษาเบสิก โดยในช่วงแรก ตัวแปลภาษาเบสิกออกแบบเป็นคอมไพเลอร์ (Compiler) ที่เป็นที่นิยมอย่างแพร่หลาย ตั้งแต่ในโรงเรียนมัธยม ไปจนถึงโรงงานอุตสาหกรรม ซึ่งใช้กฎ 8 ข้อ ในการออกแบบตัวแปลภาษาเบสิก ในยุคเริ่มต้นคือ (ธีรเดช, ม.ป.ป.)

1. ง่ายสำหรับผู้เริ่มต้นเขียนโปรแกรม
2. มีลักษณะภาษาแบบ เหมือนภาษาซี
3. สามารถเพิ่มความสามารถขั้นสูง สำหรับผู้เชี่ยวชาญได้
4. ทำงานร่วมกันได้กับผู้ใช้ (Interactive)
5. มีข้อความระบุความผิดพลาด (Error messages) ที่เข้าใจง่าย
6. ต้องทำงานอย่างรวดเร็วกับโปรแกรมขนาดเล็ก
7. ผู้ใช้ไม่ต้องเข้าใจเรื่องอุปกรณ์คอมพิวเตอร์
8. ไม่ให้ผู้ใช้ติดต่อกับระบบปฏิบัติการ

ในปี ค.ศ. 1968 เอดเจอร์ (Edsger) ได้นำเสนอวิธีการเขียนโปรแกรมอย่างเป็นระเบียบ ที่ถูกเรียกว่าการเขียนโปรแกรมแบบมีโครงสร้าง (Structured programming) ทำให้ภาษาเบสิก ที่ใช้คำสั่งโกทอ (Goto) มีโครงสร้างโปรแกรมซับซ้อนเพราะคำสั่งโกทอนุญาตให้ผู้เขียนโปรแกรมส่งถ่ายการควบคุม (Transfer of control) ไปยังจุดใด ๆ ก็ได้ในโปรแกรม ทำให้ผู้พัฒนาโปรแกรมต่อหรือผู้แก้ไขโปรแกรมไม่สามารถติดตามการทำงานของโปรแกรมได้อย่างเป็นระบบ ทำให้ภาษาเบสิกเกิดการปรับปรุงภาษาในช่วงหลัง ๆ

ในปี ค.ศ. 1975 หลังจากที่บิลเกตต์ (Bill Gates) และ พอล อัลเลน (Paul Allen) ได้สร้างตัวแปลภาษาเบสิกบนไมโครคอมพิวเตอร์สำเร็จและเจรจาการค้ากับเอ็มไอทีเอสเป็นครั้งแรก ซึ่งในปีเดียวกันนั้น เดนนิส อัลลิสัน (Dennis Allison) ได้สร้างตัวแปลภาษาเบสิกที่ชื่อ ไทน์เบสิก (Tiny BASIC) ที่คิดค้นโดยด็อกเตอร์ลีเชิน (Dr. Li-Chen) บนเครื่องที่ใช้ซีพียู 8080 สำเร็จ โดยมีบ็อบ อัลเบนท์ (Bob Albrecht) เป็นผู้สนับสนุนและให้ความช่วยเหลือ ได้เผยแพร่วิธีการสร้างและให้ซอร์ซโค้ด (Source code) ในนิตยสารที่ต่อมาเปลี่ยนชื่อเป็นวารสารของด็อกเตอร์ด็อบ (Dr. Dobb's Journal) ซึ่งเป็นนิตยสารด้านการพัฒนาซอฟต์แวร์ที่โด่งดังทั่วโลก

ณ เหตุการณ์นี้ ความพยายามในการสร้างตัวแปลภาษาเบสิกบนไมโครคอมพิวเตอร์ทำให้เกิดบุคคลสำคัญในวงการไอทีที่โด่งดังมากถึง 2 คนด้วยกัน คือบิลเกตต์จากไมโครซอฟท์และบ็อบอัลเบนท์ จากวารสารของด็อกเตอร์ด็อบและนี่อาจเป็นเหตุผลที่ไมโครซอฟท์ยังคงผลิตตัวแปลภาษาเบสิกอยู่ เพราะบิลเกตต์สร้างเนื้อสร้างตัวมาจากภาษานี้เอง

หลังจากที่ทำสัญญาขายเบสิกไปกับเครื่องแอลแทร์ (Altair) แล้วบิลเกตต์ได้ออกจากการเรียน มาตั้งบริษัทไมโครซอฟท์ ทำตัวแปลภาษาเบสิก มีขายในชื่อไมโครซอฟท์เบสิกไปยังไมโครคอมพิวเตอร์แบบอื่น ๆ ด้วยตลอดจนกลายเป็นภาษามาตรฐานบนเครื่องแอปเปิ้ลทู (Apple II) จนปี 1979 บริษัทไมโครซอฟท์ได้งานจากไอบีเอ็ม (IBM) ในการทำตัวแปลภาษาเบสิกบนเครื่องคอมพิวเตอร์ในแบบอินเทอร์พรีเตอร์ (interpreter) หรือ โปรแกรมแปลคำสั่งทำให้ในที่สุดภาษาเบสิกได้เป็นภาษามาตรฐานบน ไมโครคอมพิวเตอร์ที่ทุกเครื่องต้องมี

ในปี 1981 สถานีข่าวของอังกฤษได้จ้างให้บริษัทเอคอร์นคอมพิวเตอร์ (Acorn computers) ทำตัวแปลภาษาเบสิกในชื่อว่า บีบีซีเบสิก (BBC basic) ซึ่งพัฒนาจากอะตอมเบสิก (Atom basic) ของบริษัทเอคอร์นคอมพิวเตอร์เอง โดยเพิ่มความสามารถมากมาย เช่น การเข้าถึงระบบปฏิบัติการและการนำตัวแปลภาษาแอสเซมบลี (Assembly) มาทำร่วมด้วย เป็นต้น นอกจากนี้ นิตยสารครีเอทีฟคอมพิวเตอร์แมกกาซีน (Creative computing magazine) ก็ยังเผยแพร่ซอร์สโค้ดที่เกี่ยวกับเกมส์และอื่น ๆ อีกมากมาย

หลังจากที่เอ็มเอส ดอส (MS-Dos) ได้แพร่หลาย จากการเข้าทำตลาดของไอพีเอ็มทำให้มีการพัฒนาตัวแปลภาษาเบสิกเพิ่มขึ้น เช่น ไมโครซอฟท์ ผลิตควิกเบสิก (Quickbasic) ส่วนบอร์แลนด์ (Borland) ก็ผลิตเทอร์โบเบสิก (Turbo basic) ซึ่งพัฒนาเป็นเพอร์เวอร์เบสิก (Power basic) ทำให้ภาษาเบสิกแข่งขันพัฒนากันไปอย่างมาก มีการเพิ่มความสามารถทางด้านการจัดการเครือข่าย, ด้านกราฟฟิก, ด้านการจัดการไฟล์และเพิ่ม ประเภทของข้อมูล (Data type) ขึ้นอีก และที่สำคัญคือ การพัฒนาให้เป็นการเขียนโปรแกรมเชิงโครงสร้าง ซึ่งเป็นจุดที่โดนโจมตีมากที่สุด โดยการเพิ่มคำสั่งด้านโครงสร้างควบคุม (Control structure), โปรแกรมย่อย (Subroutine) และตัวแปรแบบโลคอล แต่ความนิยมในภาษาเบสิกก็ลดลงพร้อมกับความสำคัญของการเขียนโปรแกรม เพราะเครื่องสมัยใหม่ มักจะมาพร้อมกับโปรแกรมสำเร็จรูป ที่ผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมอีกต่อไป

ภาษาเบสิกที่ใช้ในการเขียนโปรแกรมตอนแรกเป็นแบบตัวอักษรล้วน (Text mode) ไม่มีภาพกราฟฟิกสวยงามแบบระบบวินโดวส์ อย่างไรก็ตามปัจจุบัน ต่อมาปี ค.ศ. 1990 ทางไมโครซอฟท์ ได้ประกาศเปิดตัวภาษาวิชวลเบสิก ซึ่งเป็นเหมือนกับชุดเครื่องมือ (Tool) ในการออกแบบสร้างเครื่องมือให้ติดต่อกับผู้ใช้ในรูปแบบกราฟิก (GUI: Graphic user interface) โดยใช้ภาษาเบสิกควบคุมการทำงาน จึงทำให้เป็นจุดเด่น คือผู้ที่ไม่มีพื้นฐานเรื่องการเขียนโปรแกรมก็สามารถเรียนรู้และนำไปใช้งานได้อย่างง่ายและรวดเร็ว เมื่อเทียบกับการเรียนภาษาคอมพิวเตอร์อื่น ๆ เช่น ภาษาซี ปาสคาล ฟอรัทเรน หรือ แอสเซมบลี โดยพัฒนาการของวิชวลเบสิกนั้นมาตั้งแต่เวอร์ชัน 1 ถึง 6 ซึ่งใช้งานบนระบบปฏิบัติการวินโดวส์เป็นหลัก จนปี พ.ศ. 2002 ได้เปลี่ยนเป็นวิชวลเบสิกดอทเน็ต (Visual basic.NET) ที่ทำงานบนแพลตฟอร์มแบบใหม่ของไมโครซอฟท์ที่เรียกว่า เนทเฟรมเวิร์ค (NET Framework) แล้วพัฒนาเรื่อยมาเป็นวิชวลเบสิก 2003, 2005, 2008 และวิชวลเบสิก 2010 โดยคาดว่าตัวโปรแกรมจะมีการพัฒนาต่อไปเรื่อย ๆ

ข้อดีของการเขียนโปรแกรมด้วยวิชวลเบสิก (ลักษณะ ม.ป.ป.)

สาเหตุที่วิชวลเบสิกเป็นภาษาที่เหมาะสมสำหรับการเรียนรู้ในการเขียนโปรแกรมนั้นเนื่องจากวิชวลเบสิก มีข้อดีหลายประการคือ

1. ง่ายต่อการเรียนรู้เหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องไวยากรณ์ของภาษาเองและเครื่องมือการใช้งาน
2. ความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษาเบสิกนั้นเป็นภาษาที่คนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์
3. การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต

4. ผู้พัฒนาสำคัญของวิซวลเบสิกคือบริษัทไมโครซอฟท์ซึ่งจัดว่าเป็นยักษ์ใหญ่ของวงการคอมพิวเตอร์ในปัจจุบัน เราจึงสามารถมั่นใจได้ว่าวิซวลเบสิกจะยังมีการพัฒนา ปรับปรุงและคงอยู่ไปอีกนาน

การติดตั้งโปรแกรมวิซวลเบสิกจากไมโครซอฟท์วิซวลสตูดิโอ

การติดตั้งวิซวลเบสิกจากชุดวิซวลสตูดิโอซึ่งเป็นโปรแกรมที่รวมโปรแกรมหลาย ๆ ภาษาเข้าด้วยกัน เช่น วิซวลเบสิก, วิซวลซีพลัสพลัสและ วิซวลซีชาฟนอกจากนี้ยังเป็นชุดที่สามารถใช้เครื่องมือต่าง ๆ ได้อย่างครบถ้วนสมบูรณ์ โดยความต้องการของระบบสำหรับวิซวลสตูดิโอ 2010 มีดังนี้

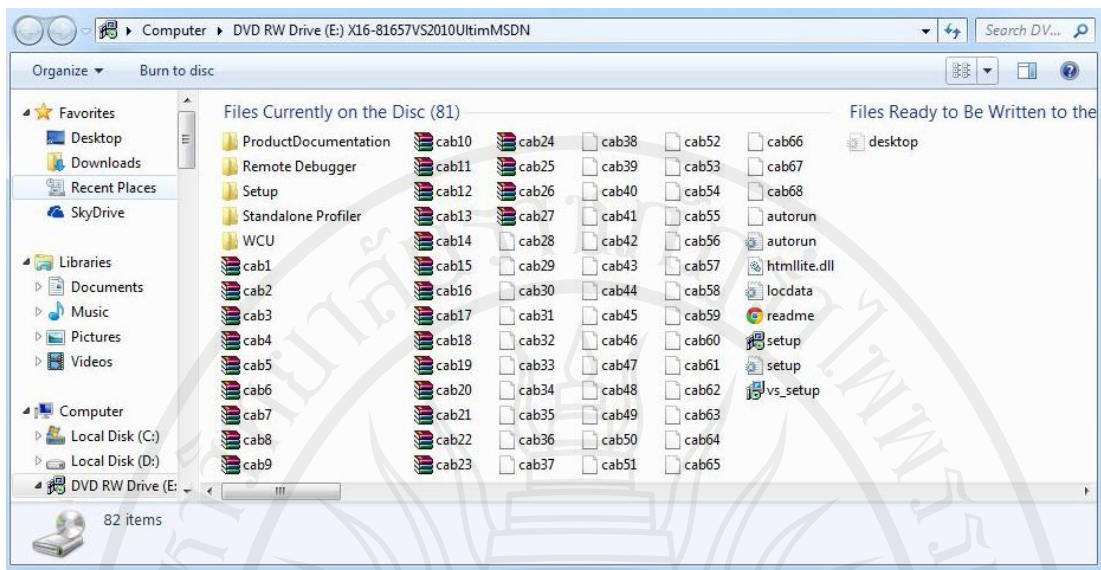
1. หน่วยประมวลผล: ไม่ควรต่ำกว่าเพนเทียม ดูอ์ คอร์ 2.20 จิกกะเฮิร์ต (Pentium Duo Core 2.20 GHz)
2. ระบบปฏิบัติการ: วินโดวส์ เอ็กซ์พี หรือ 7
3. แรม: ไม่ควรน้อยกว่า 2 จิกกะไบท์
4. พื้นที่ว่าง: ไม่น้อยกว่า 7.2 จิกกะไบท์

ขั้นตอนในการติดตั้งวิซวลสตูดิโออัลทิเมท (มงคล, ม.ป.ป.)

1. ใส่แผ่นดีวีดี ชุดการติดตั้ง วิซวลสตูดิโออัลทิเมทในไดรฟ์ดีวีดีหลังจากนั้นจะเข้าสู่การติดตั้งทันทีโดยมีให้เลือก 2 แบบคือ แบบที่ 1 ติดตั้งอัตโนมัติตั้งภาพที่ 1.39 และแบบที่ 2 แบบติดตั้งเองตั้งภาพที่ 1.40

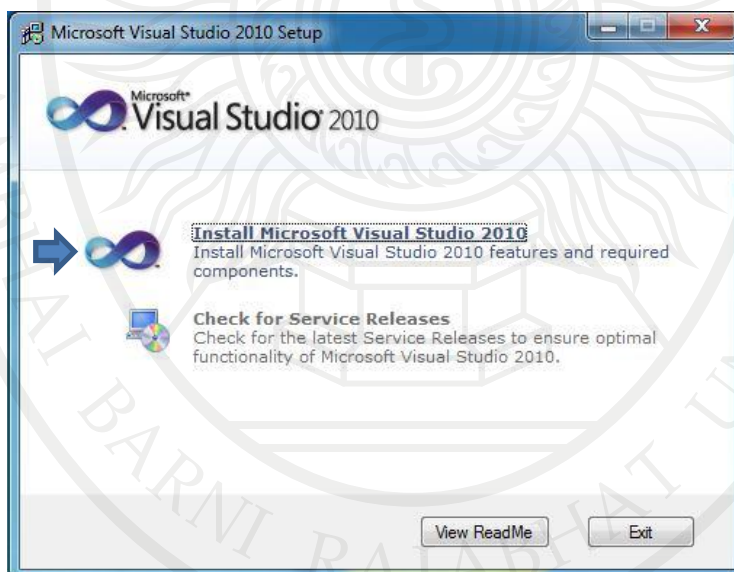


ภาพที่ 1.39 การติดตั้งวิซวลสตูดิโออัลทิเมทแบบอัตโนมัติ



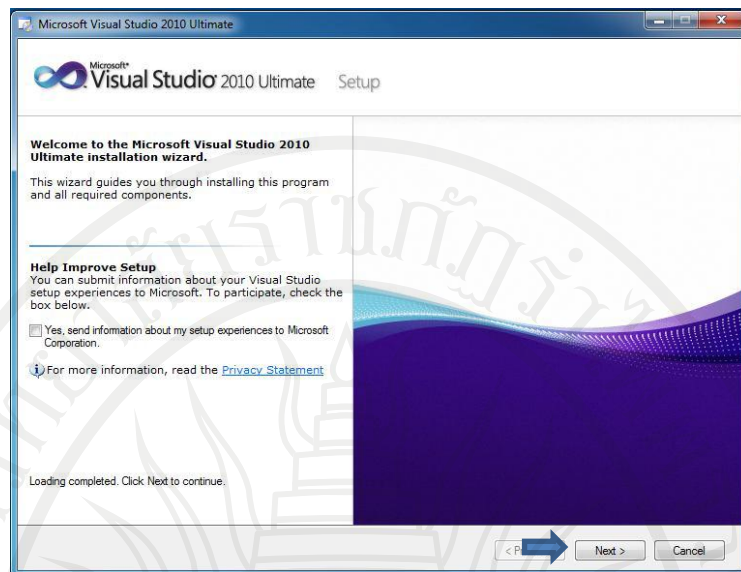
ภาพที่ 1.40 การติดตั้งวีซวลสตูดิโออัลทิเมตแบบติดตั้งเอง

2. จะปรากฏหน้าจอสำหรับติดตั้งไมโครซอฟท์วีซวลสตูดิโอ
 - 2.1 เลือกที่ “Install Microsoft Visual Studio 2010” ตามตำแหน่งดังภาพที่ 1.41



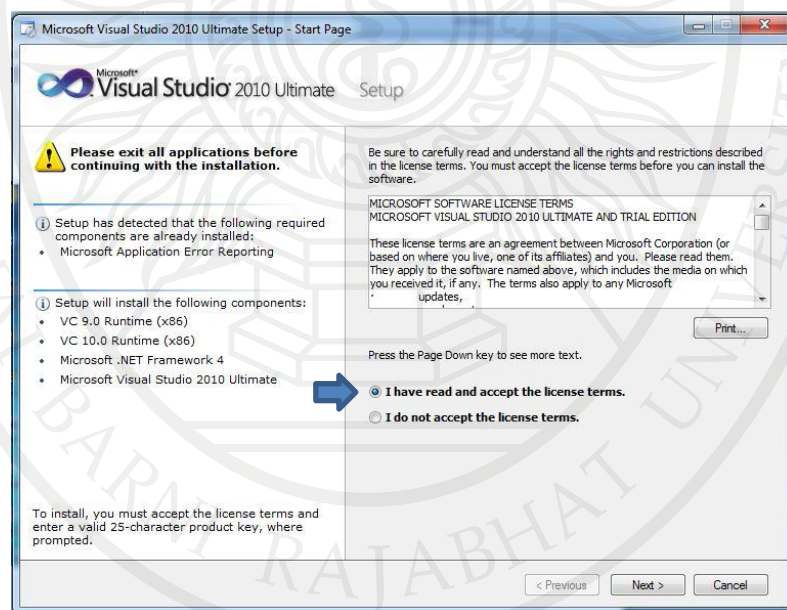
ภาพที่ 1.41 ตำแหน่งเลือกติดตั้งไมโครซอฟท์วีซวลสตูดิโอ

2.2 โปรแกรมจะทำการโหลดคอมโพเนนต์สำหรับการติดตั้ง และทำการตัดลอกไฟล์ที่จำเป็นในการใช้งาน และจะปรากฏหน้าต่าง Welcome... จากนั้นให้เลือกเครื่องหมายถูกหน้า Yes, Send Information... ออกแล้วเลือกปุ่ม Next เพื่อดำเนินการติดตั้ง ดังภาพที่ 1.42



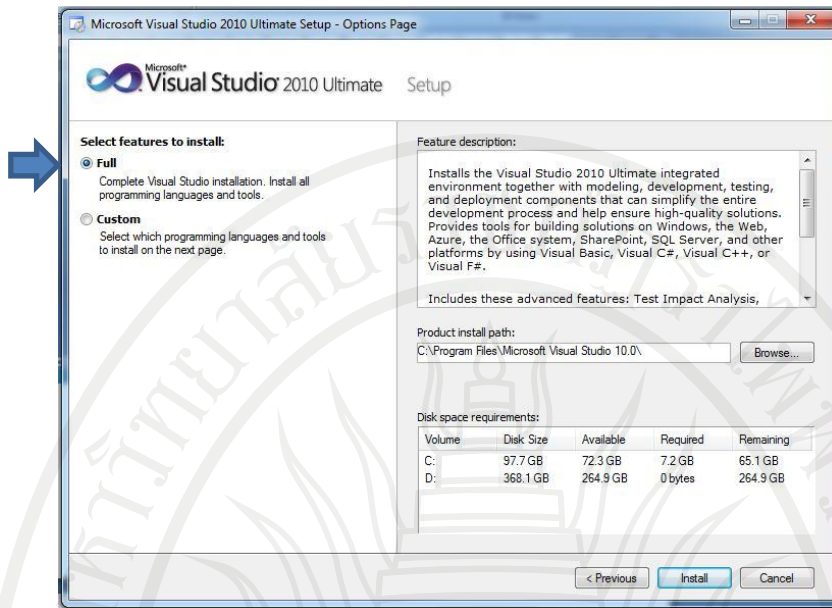
ภาพที่ 1.42 ตำแหน่งปุ่ม Next เพื่อดำเนินการติดตั้ง

2.3 เลือกเครื่องหมายที่ “I have read and accept the License terms” ดังภาพที่ 1.43 แล้วเลือกปุ่ม Next เพื่อดำเนินการต่อ



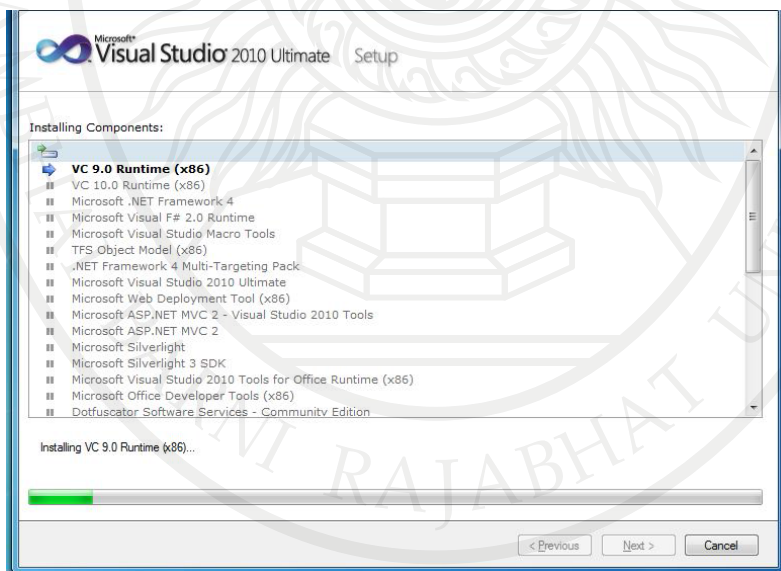
ภาพที่ 1.43 ตำแหน่งเลือกเครื่องหมายเพื่อยอมรับข้อเสนอ

3. เข้าสู่ขั้นตอนการเลือกรูปแบบการติดตั้ง ให้เลือกแบบ “Full” ดังภาพที่ 1.44 แล้วเลือก Install



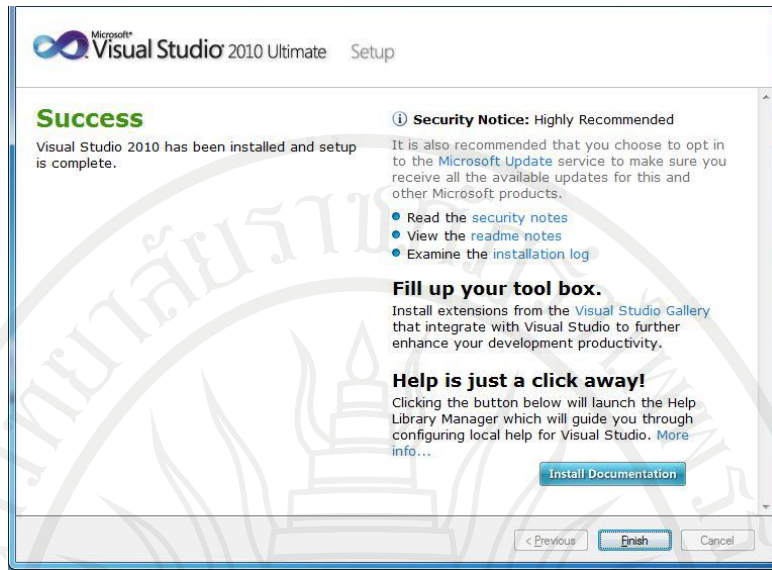
ภาพที่ 1.44 ตำแหน่งเลือกรูปแบบการติดตั้งแบบเต็มรูปแบบ

4. โปรแกรมจะทำการติดตั้งวิซวลสตูดิโอ ดังภาพที่ 1.45 ใช้เวลาประมาณ 30 นาที เมื่อติดตั้งเสร็จแล้วจะขึ้นหน้าจอนี้ ให้เลือก Finish ดังภาพที่ 1.46



ภาพที่ 1.45 โปรแกรมวิซวลสตูดิโอกำลังติดตั้ง

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 1.46 หน้าจอเมื่อติดตั้งโปรแกรมสำเร็จ

หลังจากลงโปรแกรมเป็นที่เรียบร้อยแล้ว ให้ทดลองเข้าโปรแกรมเพื่อทดสอบว่าการลงโปรแกรมนั้นเสร็จสมบูรณ์

สรุป

หลักการของคอมพิวเตอร์ ประกอบสำคัญ 4 ส่วน เริ่มจากผู้ใช้ป้อนข้อมูลผ่านอุปกรณ์ของหน่วยรับข้อมูล เช่น คีย์บอร์ด เมาส์ จากนั้นข้อมูลจึงถูกเปลี่ยนเป็นสัญญาณดิจิทัล แล้วส่งต่อไปยังหน่วยประมวลผลกลางที่คอยทำหน้าที่ควบคุมและคำนวณต่าง ๆ ตามความเร็วของซีพียู เพื่อนำไปใช้งานหรือส่งไปยังหน่วยแสดงผล เช่น จอภาพ เครื่องพิมพ์ ฯลฯ เพื่อให้แสดงผลตามความต้องการของผู้ใช้ โดยมีหน่วยความจำในแบบต่าง ๆ ไว้คอยช่วยบันทึกข้อมูลตามความต้องการหรือตามลักษณะการใช้งานทั้งในส่วนของผู้ใช้และคอมพิวเตอร์ ซึ่งในแต่ละส่วนของคอมพิวเตอร์มีอุปกรณ์ฮาร์ดแวร์ของหน่วยรับข้อมูล, อุปกรณ์ฮาร์ดแวร์ของหน่วยประมวลผล และอุปกรณ์ฮาร์ดแวร์ของหน่วยแสดงผลอีกหลากหลายรูปแบบ ซึ่งทั้งหมดได้ถูกควบคุมด้วยซอฟต์แวร์ระบบที่คอยช่วยจัดการเกี่ยวกับการดำเนินงานพื้นฐานต่าง ๆ ของคอมพิวเตอร์ ตลอดจนมีโปรแกรมที่ทำหน้าที่เป็นตัวแปลภาษาซึ่งถูกสร้างขึ้นเพื่อให้ผู้เขียนโปรแกรมเขียนชุดคำสั่ง หรือปรับปรุงแก้ไขซอฟต์แวร์ได้ง่ายในภายหลัง

ทั้งนี้แม้มีซอฟต์แวร์ใช้งานเฉพาะที่คอยช่วยจัดการและแก้ไขปัญหาต่าง ๆ แล้ว แต่อาจยังไม่เหมาะสมกับองค์กร จึงต้องมีการพัฒนาซอฟต์แวร์ให้ใช้งานเฉพาะสำหรับงานแต่ละประเภทให้ตรงกับความต้องการของผู้ใช้แต่ละราย โดยรายวิชาการโปรแกรมคอมพิวเตอร์นี้ได้ใช้โปรแกรมวิชวลเบสิก มาใช้เป็นสื่อในการเรียนการสอนเพื่อให้เกิดความรู้ความเข้าใจในการพัฒนาโปรแกรมสำหรับวิศวกรรมต่อไป

แบบฝึกหัดบทที่ 1

1. จงอธิบายหลักการทำงานของคอมพิวเตอร์
2. หน้าที่ของหน่วยความจำคืออะไร และหน่วยความจำมีกี่ชนิดอะไรบ้าง
3. อุปกรณ์ฮาร์ดแวร์แบ่งได้กี่ประเภทอะไรบ้าง
4. คีย์บอร์ดเออร์โกโนมิกส์ ช่วยลดปัญหาในการทำงานกับคอมพิวเตอร์ได้อย่างไร
5. เม้าส์แบบแสงมีหลักการทำงานแตกต่างจากเม้าส์แบบทั่วไปอย่างไร
6. แรมแบ่งออกได้กี่ประเภทอะไรบ้าง
7. อุปกรณ์แสดงผลหน้าจอมมีกี่แบบ อะไรบ้าง
8. งานเกี่ยวกับการออกใบเสร็จรับเงิน, ใบกำกับภาษี ควรใช้เครื่องพิมพ์แบบใด และเครื่องดังกล่าวมีหลักการทำงานอย่างไรบ้าง
9. เครื่องพิมพ์แบบอิงค์เจ็ทและแบบเลเซอร์ แตกต่างกันอย่างไบบ้าง จงอธิบาย
10. ตัวแปลภาษาที่เป็นที่นิยมและรู้จักกันมากในปัจจุบันมีอะไรบ้าง
11. จงเล่าประวัติโปรแกรมภาษาเบสิกมาพอสังเขป
12. จงบอกข้อดีของการเขียนโปรแกรมด้วยภาษาเบสิก

เอกสารอ้างอิง

- เกียรติศักดิ์ ทรงแก้ว. (2552). ความหมายและหลักการทำงานของ SDRAM และ DDR. [On-line] Available : <http://www.vcharkarn.com/blog/40889>. [2556, ตุลาคม 20].
- จักรกรฤษณ์ แก้วไพฑูรย์. (ม.ป.ป.). Intel core i7 โป้รเซสเซอร์สุดยอดแห่งยุค [On-line] Available : <http://www.chandra.ac.th/office/ict/project/3G%20Technology/Content6.php>. [2556, ตุลาคม 20].
- ณัฐกร สงคราม. (ม.ป.ป.). บทเรียนโปรแกรมการเรียนการสอนผ่านเว็บเรื่องคอมพิวเตอร์เบื้องต้น . [On-line] Available : http://www.thaiwbi.com/course/Intro_com/Intro_com/wbi1/hie/homepage.htm. [2556, ตุลาคม 6].
- ณรงค์ นันทวิจิตร. (2547). Mainboard. [On-line] Available : <http://www.brr.ac.th/oldweb/knowledge/hardware/mainboard/main.htm>. [2556, ตุลาคม 20].
- ธีรเดช จันทร์วิมลือง.(ม.ป.ป.). ประวัติภาษา BASIC (BASIC Programming Language History). [On-line] Available : <http://www.vb.in.th/2007/06/basic-basic-programming-language.html> [2556, ตุลาคม 6].
- นิรนาม (2555) จอภาพหรือมอนิเตอร์ (Monitor) CRT/LCD/LED คืออะไร. [On-line] Available : <http://www.thaiwebsocial.com>. [2556, ตุลาคม 20].
- นิรนาม (2556). สัญญาณอันตราย ฮาร์ดดิสก์ใกล้พัง. [On-line] Available : <http://www.rangforever.com/news.php?id=61>. [2556, ตุลาคม 20].
- มงคล มะเมียทอง. (ม.ป.ป.). เอกสารประกอบการสอน วิชาการเขียนโปรแกรมขั้นสูง ระดับชั้น ม.6. [On-line] Available : http://www.ro.ac.th/mongkolro/doc/setup_vs2010.pdf. [2556, ตุลาคม 6].
- ยงยุทธ ชมไชย. (ม.ป.ป.). หลักการทำงานของคอมพิวเตอร์. [On-line] Available : <https://sites.google.com/site/kruyutsw/3-2-hlak-kar-thangan-khxmphiwtexr>. [2556, ตุลาคม 6].
- ลักษณะ วาทิน. (ม.ป.ป.). ประวัติความเป็นมาของ VisualBasic. [On-line] Available : <https://sites.google.com/site/luxsanavathin/gui/chnid-khxm-laea-tawpaer>. [2556, ตุลาคม 6].
- วิโรจน์ ชัยมูลและสุพรรณษา ยวงทอง. (2552). ความรู้เบื้องต้นเกี่ยวกับคอมพิวเตอร์และเทคโนโลยีสารสนเทศ. กรุงเทพมหานคร: โป้รวิชัน.
- ศุภร์นภา ไกรโนนทอง. (ม.ป.ป.). กลุ่มคีย์ 5 กลุ่มบนคีย์บอร์ดมาตรฐาน. [On-line] Available : <https://sites.google.com/site/sooknapa235/keyboard> [2556, ตุลาคม 6].
- โอภาส เอี่ยมสิริวงศ์. (2547). วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ. กรุงเทพมหานคร: ซีเอ็ดยุคเคชั่น.
- Alibaba.com. (n.d.). Small Black USB Wired Optical Light Scroll Wheel Mice Mouse for PC Laptop Computer. [On-line] Available : http://www.alibaba.com/product-detail/Small-Black-USB-Wired-Optical-Light_1978012783.html. [2556, October 6].

เอกสารอ้างอิง (ต่อ)

- AltoEdge Pty. (n.d.). Microphones. [On-line] Available : <http://www.altoedge.com/microphones/>. [2556, October 6].
- Amazon.com. (n.d.). HP Wireless Elite Keyboard v2. [On-line] Available : <http://www.amazon.com/HP-Wireless-Elite-Keyboard-v2/dp/B005HNHRA6>. [2556, October 6].
- Amazon.com. (n.d.). PalmOne Portable Keyboard for Palm m100/m105, III Series, and VII Series Handhelds. [On-line] Available : <http://www.amazon.com/PalmOne-Portable-Keyboard-Palm-Handhelds/dp/B00004WHF9>. [2556, October 6].
- Amazon.com. (n.d.). Logitech Cordless Optical TrackMan. [On-line] Available : http://www.amazon.com/Logitech-Cordless-Optical-TrackMan/dp/B0032KYA6K/ref=sr_1_1?s=pc&ie=UTF8&qid=1461249934&sr=1-1 . [2556, October 6].
- Barcode118.com (2014). Barcode Scanning. [On-line] Available : <http://barcode118.com/barcode-scanning.html>. [2556, October 20].
- Bio office. (n.d.). D200 DeskTop Series FingerPrint Employee Time Clock. [On-line] Available : <http://store.bio-office.com>. [2556, October 20].
- Chatsworth (2010). ACP-2200 OMR. [On-line] Available : http://www.chatsworthdata.com/optical_mark_readers-acp_2200omr.html. [2556, October 20].
- CircuitsToday. (n.d.). Different Types Of Digital Cameras. [On-line] Available : <http://www.circuitstoday.com/different-types-of-digital-cameras>. [2556, October 6].
- Dell Inc. (n.d.). Electronic, Accessories, and Software Categories. [On-line] Available : <http://accessories.ap.dell.com/sna/default.aspx>. [2556, October 6].
- DriverBasket (n.d.). Samsung SCX-3405FW Wireless Multifunction Printer Driver. [On-line] Available : <http://driverbasket.com/samsung-scx-3405fw-printer-drivers-download/>. [2556, October 20].
- EncomPos. (n.d.). Receipt Printers. [On-line] Available : <http://encomposoftware.com/hardware.html>. [2556, October 20].
- Google Sites. (n.d.). Printer and Scanner. [On-line] Available : <https://sites.google.com/site/printerandscanner/>. [2556, October 6].
- Hearbeats (n.d.). beats-studio-red. [On-line] Available : <http://www.hearbeats.com/product/14/%E0%B8%AB%E0%B8%B9%E0%B8%9F%E0%B8%B1%E0%B8%87-beats-studio-red>. [2556, October 20].
- iPhone J.D. (2011). Review: Kensington Virtuoso Stylus for Tablet -- full length stylus for the iPad. [On-line] Available : http://www.iphonejd.com/iphone_jd/2011/08/review-kensington-virtuoso-stylus-for-tablet-full-length-stylus-for-the-ipad.html. [2556, October 6].

เอกสารอ้างอิง (ต่อ)

- Media Data Systems Pte Ltd. (n.d.). Portable Digital Video Camera with 5.0-megapixel CMOS Sensor, Supports USB 2.0 PC Interface. [On-line] Available : <http://www.globalsources.com/gsol/I/Digital-video/p/sm/1042430107.htm#1042430107>. [2556, October 6].
- Notebookspec (2011). Printer. [On-line] Available : <http://notebookspec.com/web/wp-content/uploads/2011/11/L100-550-x-3101.png>. [2556, October 20].
- OfficeMate (n.d.). Printer. [On-line] Available : <http://store.officemate.co.th/images/zpimage/4550008.jpg>. [2556, October 20].
- Pallab Chatterjee (2012). Embedded Mobile Driven by Human Interface. [On-line] Available : <http://www.digikey.com/en/articles/techzone/2012/feb/embedded-mobile-driven-by-human-interface>. [2556, October 20].
- Pelaajan Valinta. (2009). PC Pro Flight 2 joystick (PC). [On-line] Available : <http://www.pelaajanvalinta.fi/PC-Pro-Flight-2-joystick-PC>. [2556, October 6].
- Peter Vogel & Kim Ryrie. (1979). Fairlight Computer Music Instrument. [On-line] Available : <http://120years.net/fairlight-computer-music-instrument-peter-vogel-kim-ryrie-australia-1979/>. [2556, October 6].
- PortableApps.com. (2012). On-Screen Keyboard Portable. [On-line] Available : http://portableapps.com/apps/accessibility/on-screen_keyboard_portable. [2556, October 6].
- Redmond Pie. (n.d.). Future Notebooks Will Have Fingerprint Scanners Built Right Into The Touchpad. [On-line] Available : <http://www.redmondpie.com/future-notebooks-will-have-fingerprint-scanners-built-right-into-the-touchpad/>. [2556, October 6].
- Sajee (2013). speaker. [On-line] Available : <https://sajee1234.files.wordpress.com/2013/07/81934.jpg>. [2556, October 20].
- SlashGea. (n.d.). LG ET83 touchscreen monitor wants your Windows 8 fingers. [On-line] Available : <http://www.slashgear.com/lg-et83-touchscreen-monitor-wants-your-windows-8-fingers-31254980/>. [2556, October 6].
- SV Plannet Ltd (n.d.). Projector. [On-line] Available : <http://www.svplannet.com/Projector>. [2556, October 20].
- Truly Ergonomic. (n.d.). The Truly Ergonomic Mechanical Keyboard. [On-line] Available : <https://www.trulyergonomic.com/store/index.php>. [2556, October 6].
- Weebly. (n.d.). Digitizing Tablet. [On-line] Available : <http://shaniqueportfolio.weebly.com/digitizing-tablet.html>. [2556, October 6].

เอกสารอ้างอิง (ต่อ)

Ziff Davis. (n.d.). Logitech Webcam c930e. [On-line] Available : <http://www.pcmag.com/article2/0,2817,2423466,00.asp>. [2556, October 6].



ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

แผนบริหารการสอนประจำบทที่ 2

หัวข้อเนื้อหาบทที่ 2

1. หลักการเขียนผังงานเบื้องต้น
2. การเขียนผังงานแบบลำดับ
3. การเขียนผังงานแบบเลือกเส้นทาง
4. การเขียนผังงานแบบทำซ้ำ
5. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 2 แล้วนักเรียนสามารถ

1. รู้หลักการเขียนผังงานเบื้องต้น
2. รู้จักและสามารถเขียนผังงานแบบลำดับได้
3. รู้จักและสามารถเขียนผังงานแบบเลือกเส้นทางได้
4. รู้จักและสามารถเขียนผังงานแบบทำซ้ำได้
5. สรุป

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายเนื้อหา
2. ให้นักศึกษาได้เรียนรู้ผ่านการบรรยายที่จอโปรเจคเตอร์
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 2

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
4. แบบฝึกหัดบทที่ 2

การวัดผลและประเมินผล

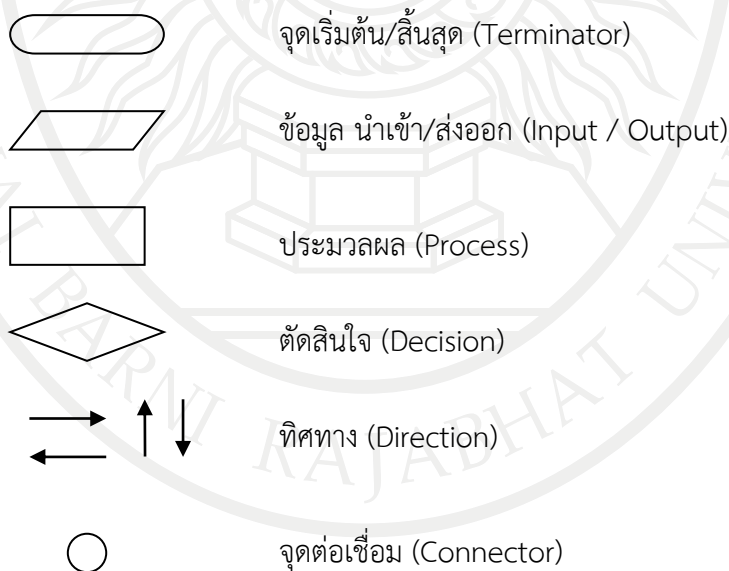
1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากกิจกรรมร่วม
3. สังเกตจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด

บทที่ 2 การเขียนผังงาน

การเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาคอมพิวเตอร์ระดับสูงเพื่อนำไปใช้งาน จำเป็นต้องมีอัลกอริทึม (Algorithm) หรือกระบวนการแก้ปัญหาที่สามารถเข้าใจได้อย่างมีลำดับหรือวิธีการในการแก้ไขปัญหาใดปัญหาหนึ่งอย่างเป็นขั้นเป็นตอนและชัดเจน (วิโรจน์ และสุพรรณษา, 2552) ซึ่งการเขียนผังงานนั้นเป็นเทคนิคหรือวิธีการอย่างหนึ่งสำหรับใช้เขียนแสดงกระบวนการทางความคิดในการแก้ปัญหา เพื่อให้เกิดความเข้าใจและไม่หลงทางในเวลาพัฒนาโปรแกรม ทั้งนี้เพื่อให้ผู้อื่นได้เข้าใจในสิ่งที่ผู้เขียนโปรแกรมคิด การเขียนผังงานจึงต้องมีการศึกษาเพื่อให้อ่านรูปร่างที่เป็นสัญลักษณ์มาตรฐานสากลเขียนแทนกิจกรรมต่าง ๆ โดยสัญลักษณ์ภาพนี้กำหนดตามมาตรฐานของเอเอ็นเอสไอ (ANSI: American national standards institute) และไอโอเอส (ISO: International standard organization) หากนักศึกษาหรือนักพัฒนาโปรแกรมได้ศึกษาเนื้อหาการเขียนผังงานจะช่วยให้เกิดการพัฒนาทางความคิดอย่างเป็นระบบและยังสามารถทำให้ผู้ที่สนใจอื่น ๆ เข้าใจความคิดของผู้เขียนโปรแกรมได้ง่าย ตลอดจนยังสามารถทำให้เข้าใจเนื้อหาในบทต่อไปได้ง่ายยิ่งขึ้นด้วย

หลักการเขียนผังงานเบื้องต้น

ผังงาน (Flowchart) คือแผนภาพที่มีขั้นตอนของคำสั่งการทำงานโดยใช้สัญลักษณ์ในการเขียน ซึ่งเป็นสัญลักษณ์รูปร่างและลูกศรที่แสดงถึงขั้นตอน ทิศทางการไหลของข้อมูลที่ขั้นตอนตั้งแต่แรกจนได้ผลลัพธ์ตามที่ต้องการ โดยการเขียนผังงานเบื้องต้นสามารถใช้สัญลักษณ์ดังนี้ (สุรัตน์, ม.ป.ป.)



หลักการใช้สัญลักษณ์เบื้องต้น มีดังนี้

1. ต้องมีจุดเริ่มต้น (Start) และจุดสิ้นสุด (End)
2. สัญลักษณ์ต้องมีการเชื่อมต่อกันด้วย ทิศทางการทำงาน (Direction of flow)
3. ทิศทางการทำงานจะต้องเริ่มที่จุดเริ่มต้นและจบที่จุดสิ้นสุดเท่านั้น

นอกจากสัญลักษณ์ในการเขียนผังงานแล้ว การเขียนผังงานยังจำเป็นต้องรู้อัลกอริทึมที่เป็นวิธีการในการแก้ไขปัญหาใดปัญหาหนึ่งอย่างเป็นขั้นเป็นตอนและชัดเจนด้วย เช่น เมื่อต้องการเดินทางไปมหาวิทยาลัย ต้องทำอะไรบ้าง

วิธีที่ 1

- 1.นั่งรถสองแถวจากบ้านไปลงตลาด
- 2.นั่งรถสองแถวจากตลาดไปมหาวิทยาลัย
- 3.ถึงมหาวิทยาลัย

วิธีที่ 2

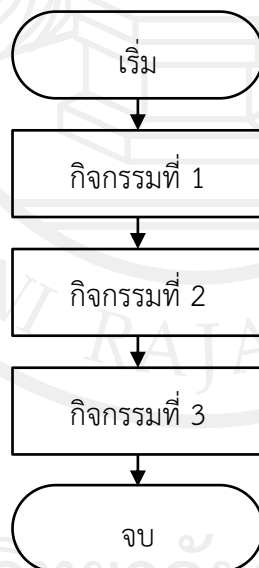
- 1.ดูสภาพอากาศ
 - ถ้าอากาศดีขี่มอเตอร์ไซด์
 - ถ้าอากาศไม่ดีขี่รถยนต์
- 2.ถึงมหาวิทยาลัย

ซึ่งสังเกตได้ว่าใน 1 ปัญหา มีกระบวนการและวิธีแก้ไขหลายวิธี โดยแต่ละคนอาจมีวิธีแก้ไขปัญหาที่แตกต่างกันออกไป จากนั้นเมื่อรู้จักอัลกอริทึมแล้ว ให้นำอัลกอริทึมที่คิดได้ ไปเขียนเป็นผังงานตามลำดับขั้นตอนของอัลกอริทึมต่อไป

โครงสร้างควบคุมพื้นฐานของผังงานทำหน้าที่ควบคุมการทำงานของโปรแกรมว่าไปในทิศทางใดนั้นสามารถแบ่งออกเป็น 3 ประเภท ดังนี้ (ไอบาส, 2547)

การเขียนผังงานแบบตามลำดับ

การเขียนผังงานแบบตามลำดับ (Sequence) เป็นโครงสร้างควบคุมการทำงานที่ละคำสั่งจากบนลงล่างหรือตามลำดับก่อนหลัง ตามทิศทางการทำงานดังภาพที่ 2.1



ภาพที่ 2.1 รูปแบบผังงานแบบตามลำดับ

คำบรรยายรูปแบบผังงานแบบตามลำดับ



ตัวอย่างที่ 2.1 การเขียนผังงานขั้นตอนการส่งจดหมาย
อัลกอริทึม

1. จ่าหน้าซองจดหมาย
2. พับจดหมาย
3. ใส่ซองจดหมาย
4. ปิดผนึกซองจดหมาย
5. ตัดแสตมป์
6. ใส่ตู้จดหมาย

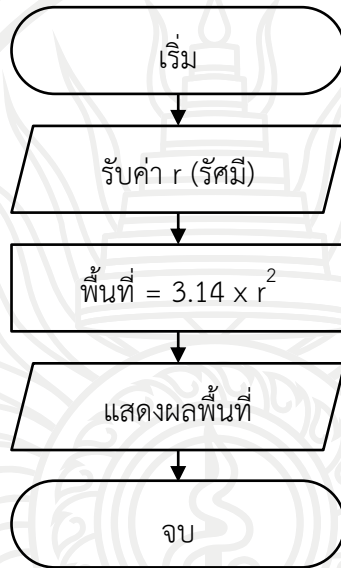
ผังงาน



ตัวอย่างที่ 2.2 การเขียนผังงานขั้นตอนการคำนวณหาพื้นที่วงกลม
อัลกอริทึม

1. รับค่า r (รัศมี)
2. คำนวณตามสูตร พื้นที่ = $3.14 \times r^2$
3. แสดงผล

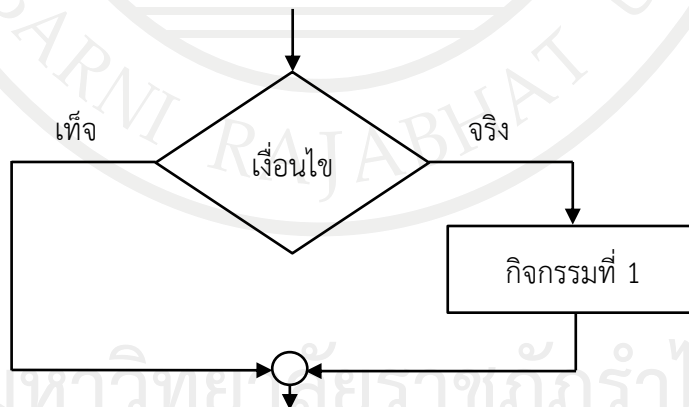
ผังงาน



การเขียนผังงานแบบเลือกเส้นทาง

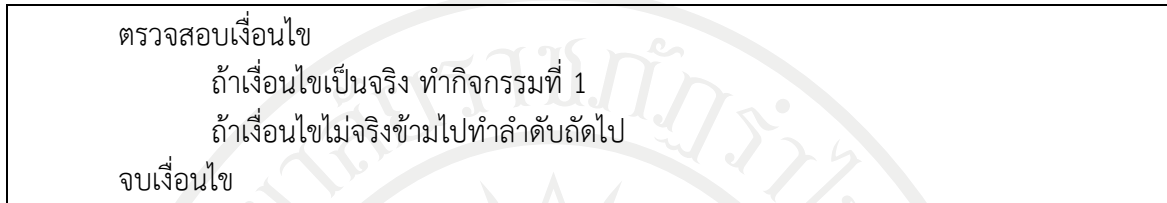
การเขียนผังงานแบบเลือกเส้นทาง (Selection) เป็นโครงสร้างการทำงานที่มีการตัดสินใจโดยมีเงื่อนไขเป็นตัวกำหนด เมื่อเลือกเส้นทางแล้วไม่ว่าจะเป็นเส้นทางใดต้องมาพบกันและทำงานในขั้นตอนต่อไป โดยผังงานแบบเลือกเส้นทางแบ่งออกเป็น 3 รูปแบบด้วยกัน คือ (โอภาส, 2547)

ผังงานแบบเลือกเส้นทางรูปแบบที่ 1 การเลือกทำตามคำสั่งเมื่อตรวจสอบเงื่อนไขว่าเป็นจริงเท่านั้น

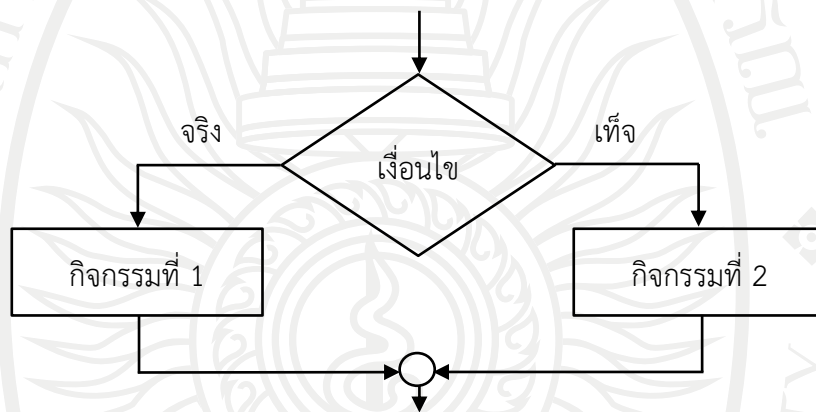


ภาพที่ 2.2 ผังงานแบบเลือกเส้นทางรูปแบบที่ 1

คำบรรยายรูปแบบจากภาพที่ 2.2 การเลือกทำตามคำสั่งเมื่อตรวจสอบเงื่อนไขว่าเป็นจริงเท่านั้น

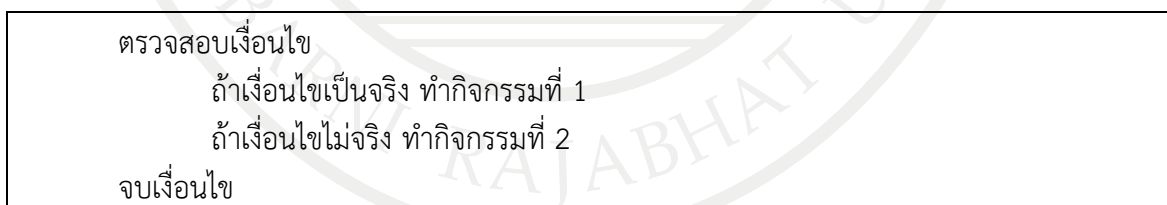


ผังงานแบบเลือกเส้นทางรูปแบบที่ 2 การเลือกทำตามคำสั่งใดคำสั่งหนึ่งเท่านั้น โดยตรวจสอบเงื่อนไขว่าเป็นจริงหรือเท็จ



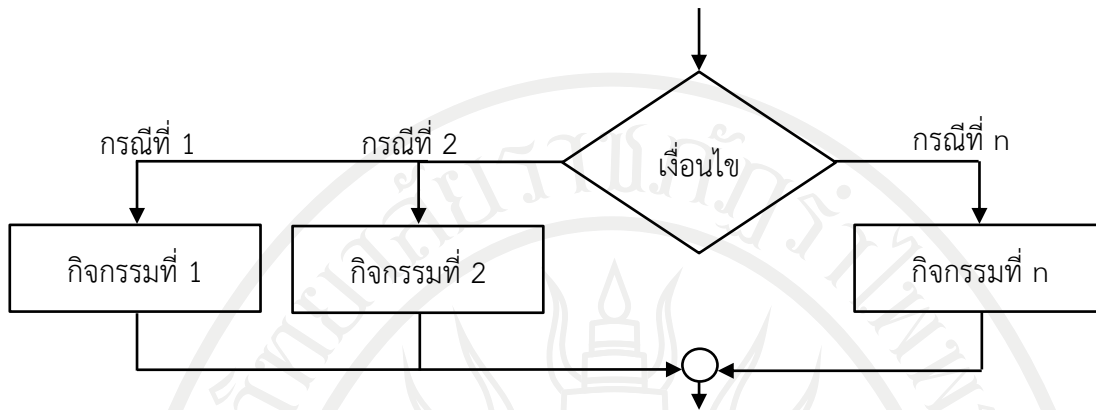
ภาพที่ 2.3 ผังงานแบบเลือกเส้นทางรูปแบบที่ 2

คำบรรยายผังงานจากภาพที่ 2.3 การเลือกทำตามคำสั่งใดคำสั่งหนึ่งเท่านั้น โดยตรวจสอบเงื่อนไขว่าเป็นจริงหรือเท็จ



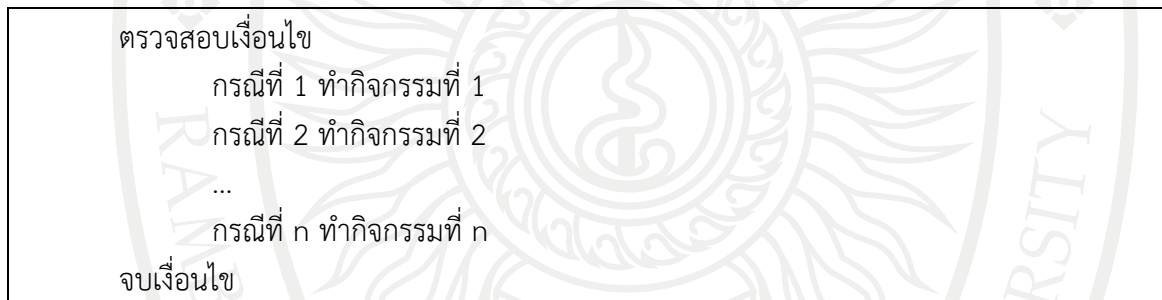
ผังงานแบบเลือกเส้นทางรูปแบบที่ 3 เป็นคำสั่งในการเลือกทำตามคำสั่งใดคำสั่งหนึ่งเท่านั้น โดยตรวจสอบค่าข้อมูลที่นำมาเปรียบเทียบกับเงื่อนไขใด ถ้าเป็นจริงก็จะเข้ากรณีนั้น

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 2.4 ผังงานแบบเลือกเส้นทางรูปแบบที่ 3

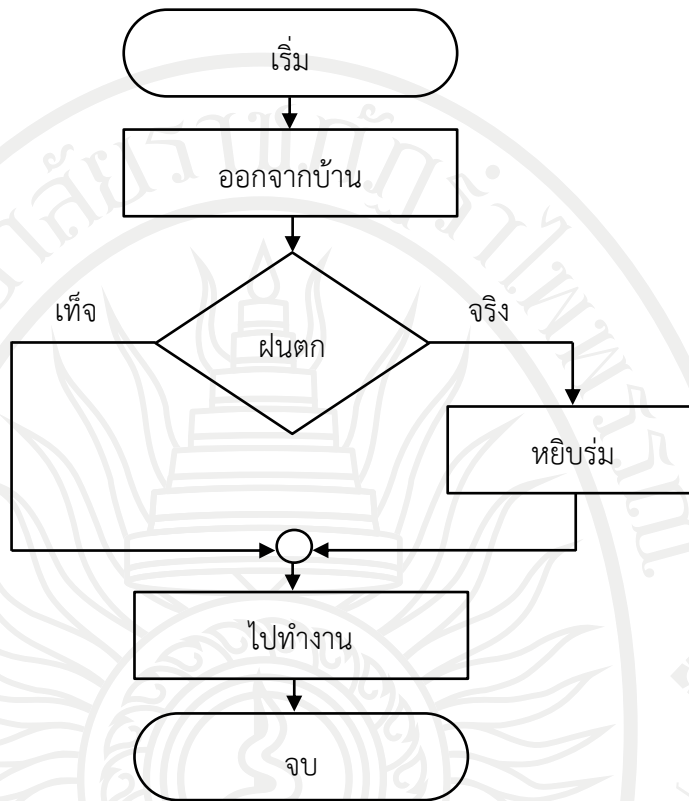
คำบรรยายรูปแบบจากภาพที่ 2.4 การเลือกทำคำสั่งใดคำสั่งหนึ่งเท่านั้น โดยตรวจสอบค่าข้อมูลที่นำมาเปรียบเทียบตรงกับเงื่อนไขใด ถ้าเป็นจริงก็ให้เข้ากรณีนั้น



ตัวอย่างที่ 2.3 การเขียนผังงานก่อนออกจากบ้านไปทำงาน
อัลกอริทึม

1. ออกจากบ้าน
2. ตรวจสอบเงื่อนไขว่า ถ้าฝนตก
เงื่อนไขเป็นจริง ให้หยิบร่ม
3. ไปทำงาน

ผังงาน



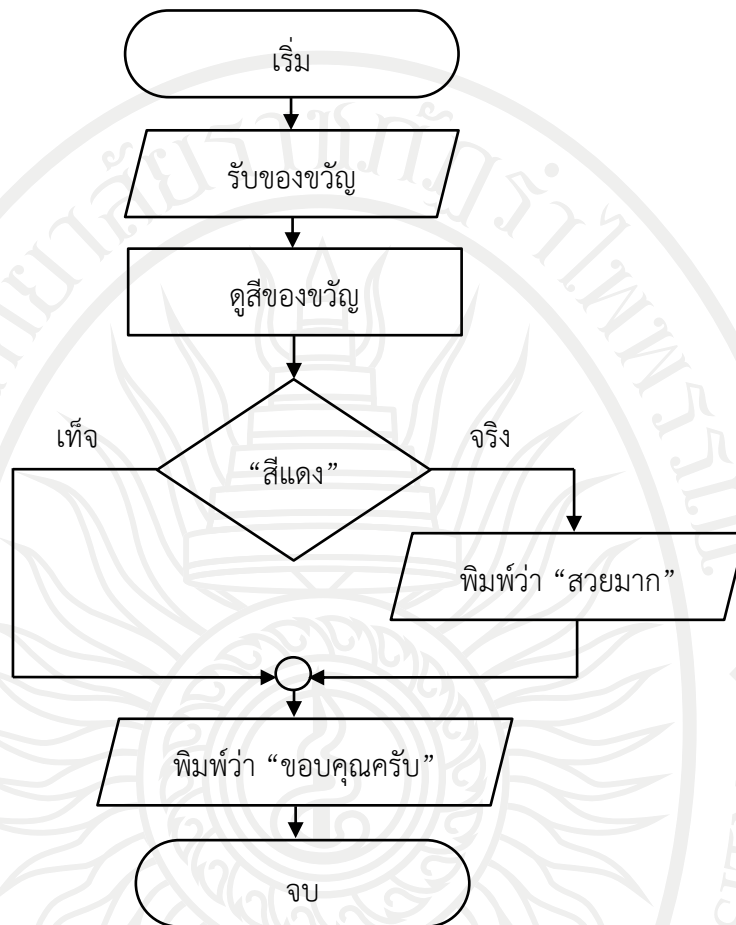
จากรูปแบบของผังงานพบว่า เมื่อออกจากบ้าน ไม่ว่าจะฝนจะตกหรือไม่ ยังไงก็ต้องออกไปทำงานแน่นอน หากแต่ถ้าถ้าฝนตกให้หยุดไปด้วย ผังงานแบบนี้จึงเป็นรูปแบบผังงานในการเลือกทำตามคำสั่ง เมื่อตรวจสอบเงื่อนไขว่าเป็นจริงเท่านั้น

ตัวอย่างที่ 2.4 การเขียนผังงาน การรับของขวัญเมื่อได้ของที่ถูกต้อง
อัลกอริทึม

1. รับของขวัญ
2. ดูสีของขวัญ
3. ตรวจสอบเงื่อนไขว่า ถ้ารับค่าเป็น “สีแดง”
เงื่อนไขเป็นจริงให้พิมพ์ชมว่า “สวยมาก”
4. พิมพ์ว่า “ขอบคุณครับ”

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ผังงาน



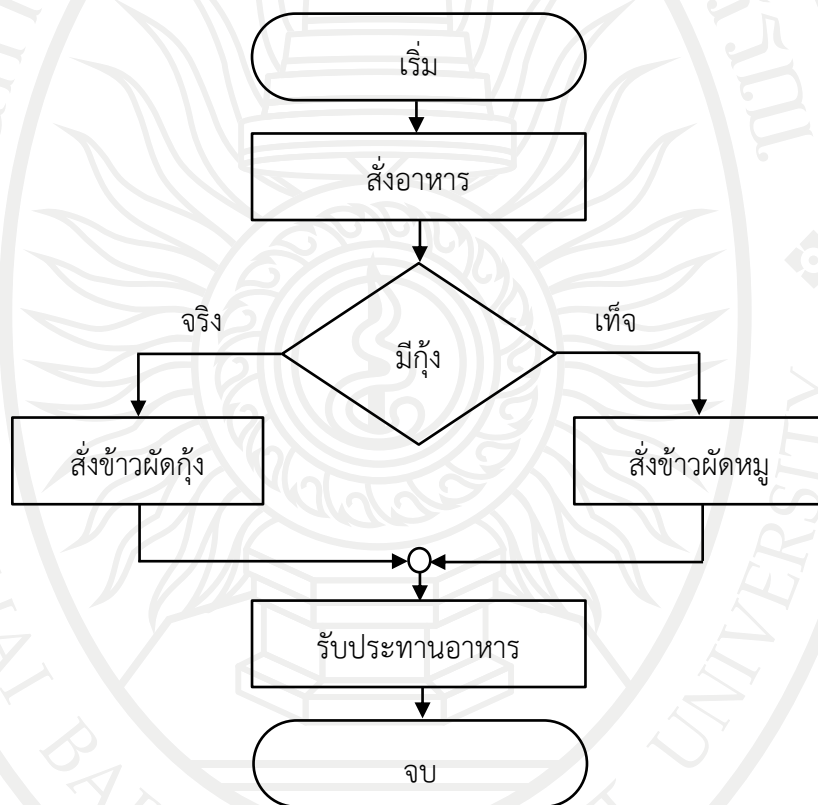
จากรูปแบบของผังงานพบว่า การรับของขวัญใช้สัญลักษณ์เป็นสี่เหลี่ยมขนมเปียกปูนเพื่อแสดงถึงข้อมูลนำเข้า จากนั้นดูสีของขวัญใช้สัญลักษณ์เป็นสี่เหลี่ยมผืนผ้าเพื่อแสดงถึงการประมวลผล และทำการตรวจสอบเงื่อนไข ซึ่งไม่ว่าจะได้สีอะไรก็ตาม จะพิมพ์คำว่า “ขอบคุณครับ” เสมอ หากเพียงแต่ถ้าเป็นสีแดงจะพิมพ์คำว่า “สวยงาม” ก่อนแล้วค่อยพิมพ์คำว่า “ขอบคุณครับ” โดยในส่วนที่เป็นการพิมพ์นี้ใช้สัญลักษณ์เป็นสี่เหลี่ยมขนมเปียกปูนเพื่อแสดงถึงข้อมูลส่งออก

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ตัวอย่างที่ 2.5 การเขียนผังงานสั่งอาหารในร้านอาหารแบบที่ 1
อัลกอริทึม

1. สั่งอาหาร
2. ตรวจสอบเงื่อนไขว่า ที่ร้านมีก๋วย
เจ๋นไช่เป็นจริง สั่งข้าวผัดก๋วย
เจ๋นไช่เป็นเท็จ สั่งข้าวผัดหมู
3. รับประทานอาหาร

ผังงาน

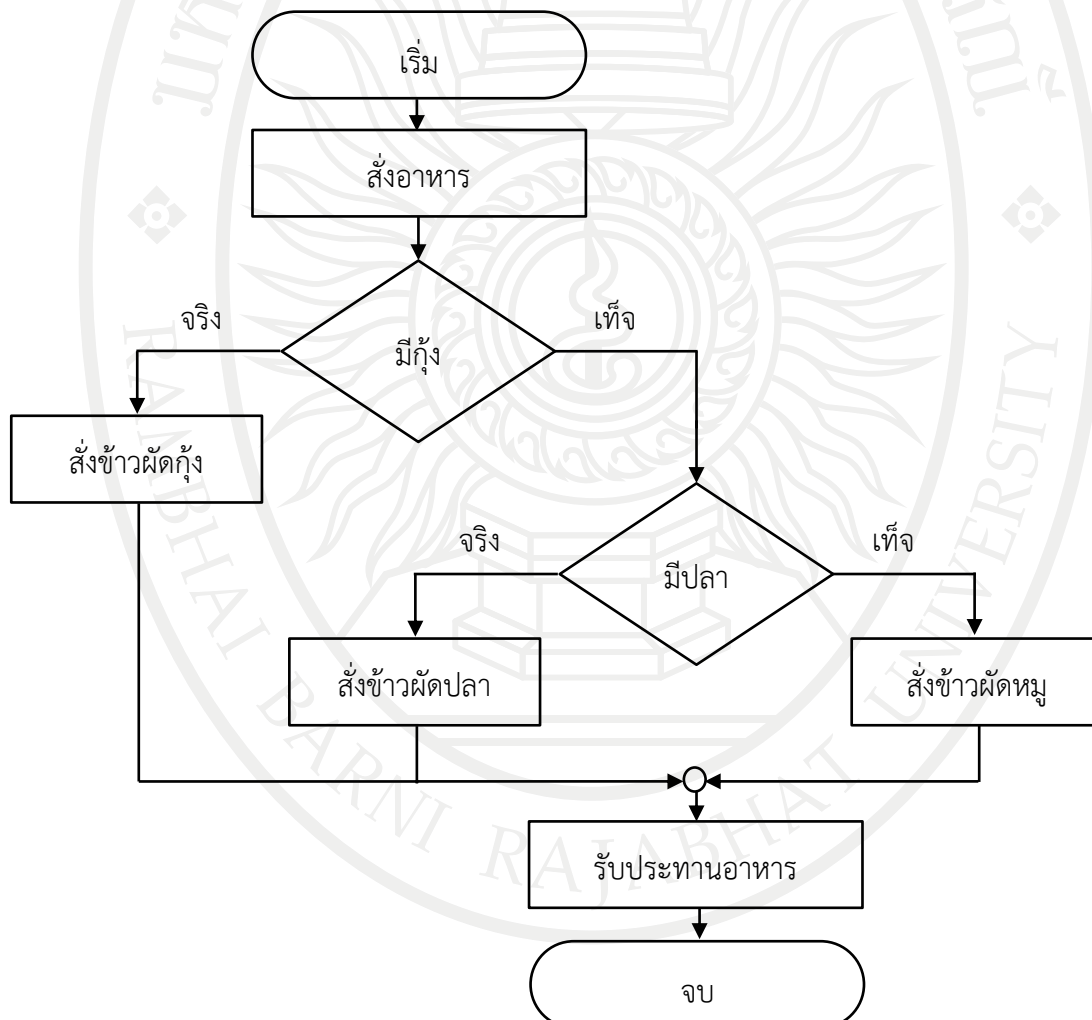


จากรูปแบบของผังงานจะพบว่า เมื่อสั่งอาหารแล้ว เงื่อนไขการตัดสินใจต่อมาอยู่ที่ทางร้านมีอาหารที่เราต้องการหรือไม่ ซึ่งในกรณีนี้คือ ทางร้านมีก๋วยหรือไม่ ถ้ามีแสดงว่าเงื่อนไขเป็นจริง ให้สั่งข้าวผัดก๋วย ส่วนถ้าไม่มีคือเป็นเท็จ ให้สั่งข้าวผัดหมูแทน ไม่ว่าจะเงื่อนไขจะเป็นจริงหรือเป็นเท็จก็จะเข้าสู่กระบวนการถัดไปคือการรับประทานอาหาร เป็นอันจบโปรแกรม

ตัวอย่างที่ 2.6 การเขียนผังงานสั่งอาหารในร้านอาหารแบบที่ 2 (มีการตรวจสอบเงื่อนไขซ้อนเงื่อนไข)
อัลกอริทึม

1. สั่งอาหาร
2. ตรวจสอบเงื่อนไขว่า ที่ร้านมีก๋วย
เงื่อนไขเป็นจริง สั่งข้าวผัดก๋วย
เงื่อนไขเป็นเท็จ ตรวจสอบเงื่อนไขว่า ที่ร้านมีปลา
เงื่อนไขเป็นจริง สั่งข้าวผัดปลา
เงื่อนไขเป็นเท็จ สั่งข้าวผัดหมู
3. รับประทานอาหาร

ผังงาน

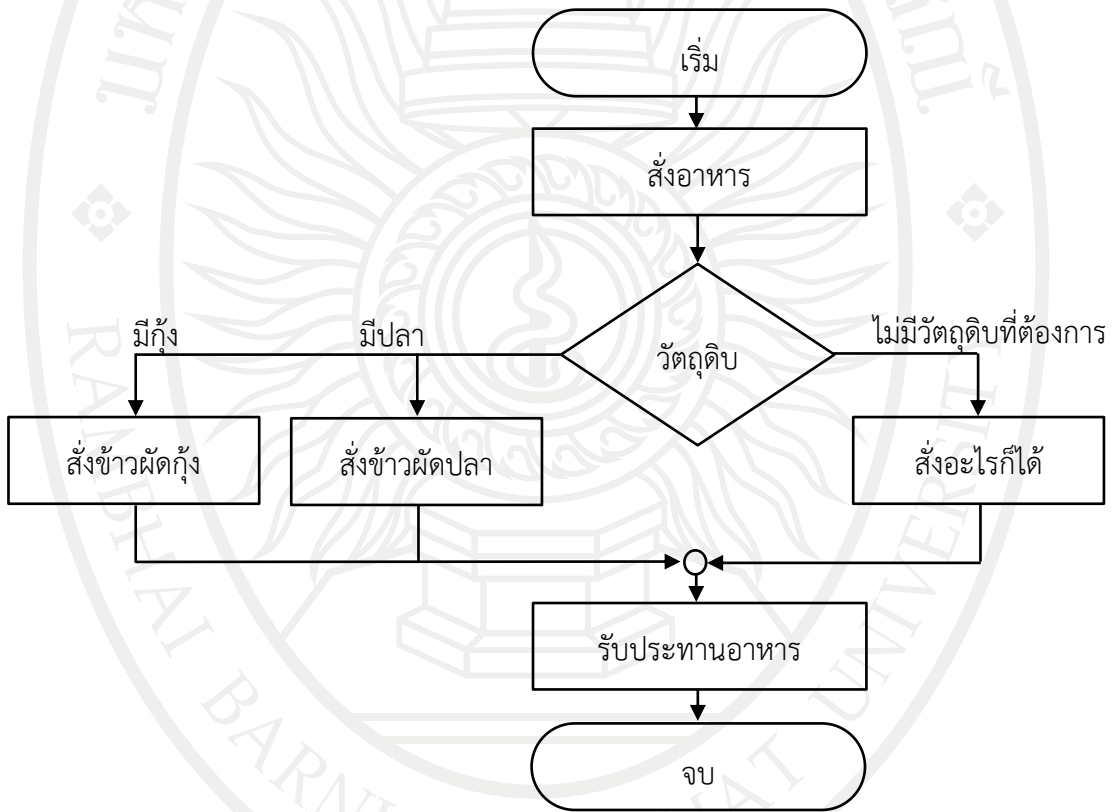


จากรูปแบบของผังงานพบว่าสามารถมีเงื่อนไขที่ซ้อนเงื่อนไขเข้าไปได้อีกเพื่อให้เกิดโอกาสในการเลือกตัดสินใจได้มากขึ้น

ตัวอย่างที่ 2.7 การเขียนผังงานการสั่งอาหารในร้านอาหารแบบที่ 3
อัลกอริทึม

- 1. สั่งอาหาร
- 2. ตรวจสอบเงื่อนไข
 - มีก๋วย
 - มีปลา
 - ไม่มีวัตถุดิบที่ต้องการ
- 3. รับประทานอาหาร

ผังงาน



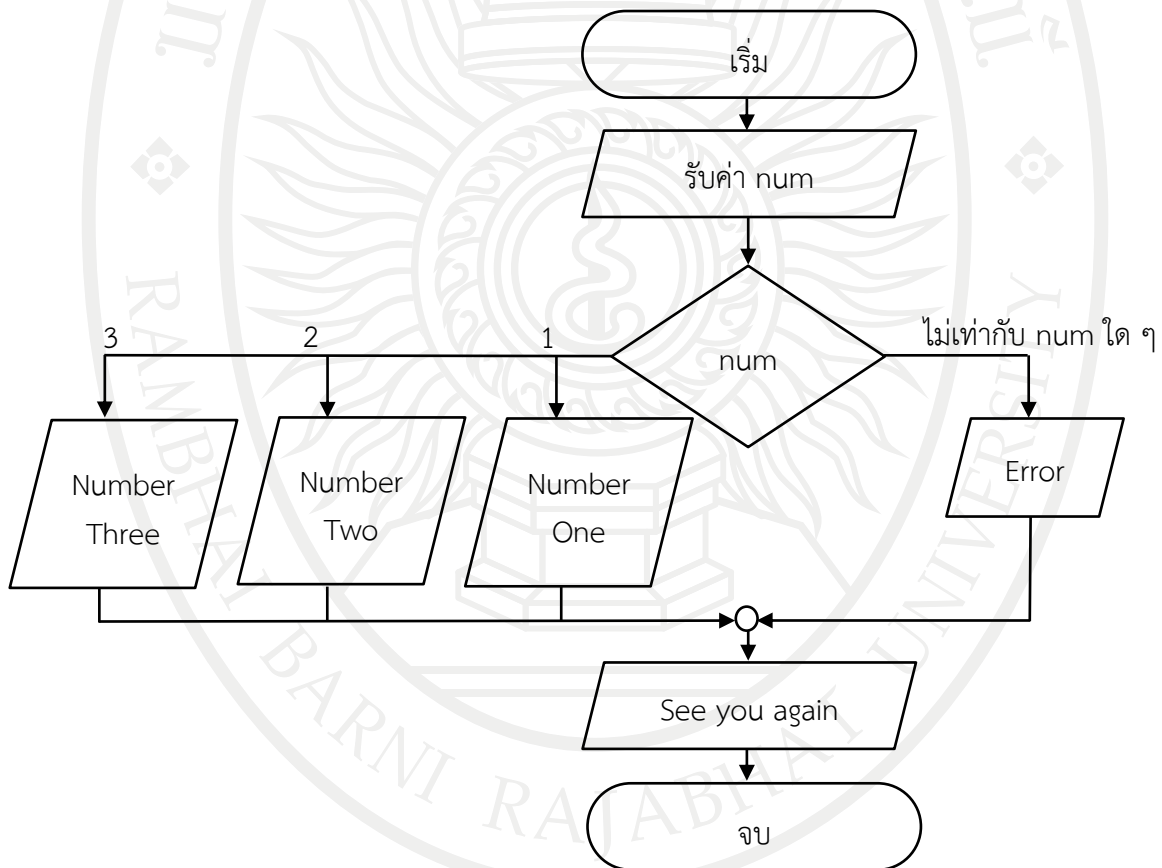
ในกรณีรูปแบบผังงานแบบเลือกเส้นทาง จากตัวอย่างมีด้วยกัน 3 กรณีคือถ้าวัตถุดิบทางร้านมี ก๋วย มีปลา และไม่มีวัตถุดิบที่ต้องการ ซึ่งเราสามารถเพิ่มกรณีเข้าไปอีกได้เช่น มีหมึก มีหมู มีไก่ ตามแต่ โจทย์ของเรา เมื่อเลือกตรงตามวัตถุดิบก็จะผ่านกระบวนการต่อไปจนถึงกระบวนการรับประทานอาหาร แต่ ถ้าไม่มีวัตถุดิบที่ต้องการก็จะสั่งเป็นอะไรก็ได้แทน แล้วค่อยรับประทานอาหาร โดยในโจทย์การสั่งอาหารนี้เป็นการสั่งเพื่อเลือกวัตถุดิบเพียงแค่ครั้งเดียว ซึ่งอาจต้องใช้ผังงานแบบทำซ้ำเข้ามาช่วยด้วย เพราะในชีวิตจริงเราอาจเลือกมากกว่านั้นแล้วค่อยรับประทานอาหาร

ตัวอย่างที่ 2.8 การเขียนผังงานการแสดงผลตัวเลข

อัลกอริทึม

1. รับค่า num
2. ตรวจสอบเงื่อนไข
 - ถ้า num=1 แสดงข้อความ "Number One"
 - ถ้า num=2 แสดงข้อความ "Number Two"
 - ถ้า num=3 แสดงข้อความ "Number Three"
 - ถ้า ค่าไม่เท่ากับ num แสดงข้อความ "Error"
3. แสดงข้อความ "See you again"

ผังงาน

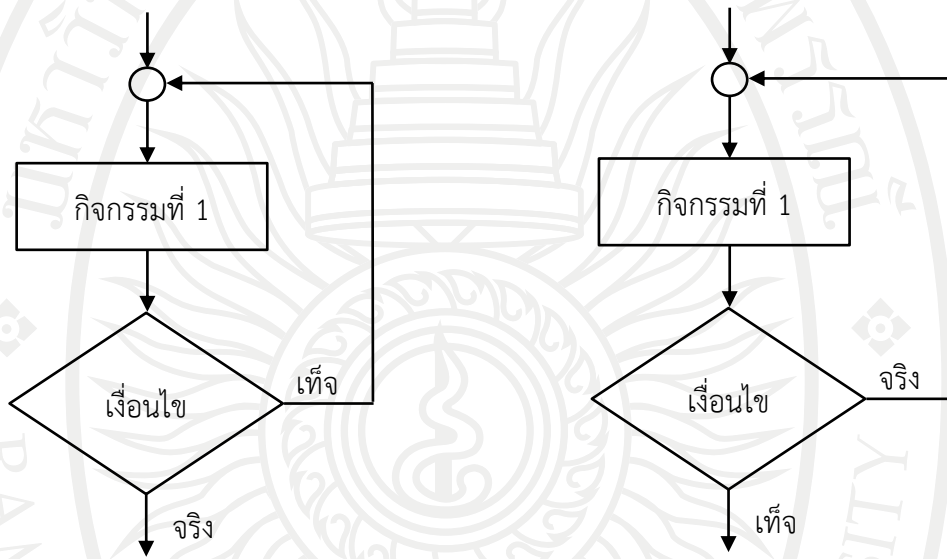


ในกรณีรูปแบบผังงานแบบเลือกเส้นทาง เมื่อนำไปใช้จริงจะมีการรับค่า ซึ่งใช้สัญลักษณ์เป็นสี่เหลี่ยมขนมเปียกปูน ทั้งในส่วนที่เป็นข้อมูลนำเข้าและแสดงผล

การเขียนผังงานแบบทำซ้ำ

การเขียนผังงานแบบทำซ้ำ (Iteration) เป็นโครงสร้างที่มีการทำงานแบบทำซ้ำคำสั่ง คือให้กลับมาทำงานในขั้นตอนอย่างเก่า ตามเงื่อนไขที่กำหนด ซึ่งลักษณะการวนนี้เรียกว่า Loop แบ่งออกเป็น 2 รูปแบบด้วยกัน คือ (โอภาส, 2547)

ผังงานแบบทำซ้ำรูปแบบที่ 1 จะทำตามคำสั่งก่อน 1 ครั้ง แล้วค่อยตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริง ให้กลับไปทำคำสั่งอีกครั้ง จนกว่าเงื่อนไขที่ตรวจสอบนั้นเป็นเท็จ หรือทำตามคำสั่งก่อน 1 ครั้ง แล้วค่อยตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นเท็จให้กลับไปทำคำสั่งอีกครั้ง จนกว่าเงื่อนไขที่ตรวจสอบนั้นจะเป็นจริง

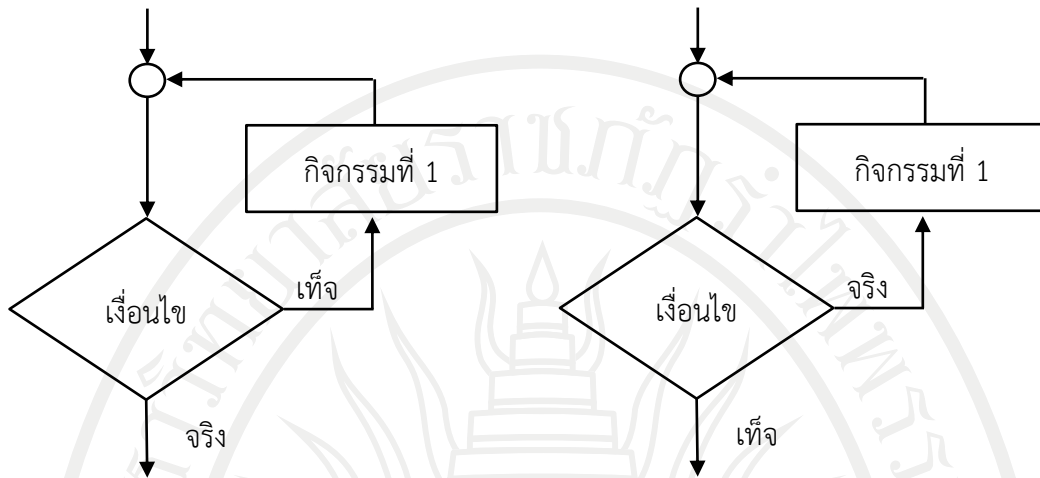


ภาพที่ 2.5 ผังงานแบบทำซ้ำรูปแบบที่ 1

คำบรรยายรูปแบบจากภาพที่ 2.5 ผังงานแบบทำซ้ำรูปแบบที่ 1

กิจกรรมที่ 1
ทำซ้ำจนกระทั่งเงื่อนไขเป็นจริงหรือเท็จ
จบการทำซ้ำ

ผังงานแบบทำซ้ำรูปแบบที่ 2 เป็นคำสั่งในการตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็นจริง ให้ทำตามคำสั่งถัดไปแล้วกลับไปตรวจสอบเงื่อนไขอีกครั้ง จนกว่าเงื่อนไขที่ตรวจสอบนั้นเป็นเท็จ หรือเป็นคำสั่งในการตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็นเท็จ ให้ทำตามคำสั่งถัดไปแล้วกลับไปตรวจสอบเงื่อนไขอีกครั้ง จนกว่าเงื่อนไขที่ตรวจสอบนั้นเป็นจริง



ภาพที่ 2.6 ผังงานแบบทำซ้ำรูปแบบที่ 2

คำบรรยายรูปแบบจากภาพที่ 2.6 ผังงานแบบทำซ้ำรูปแบบที่ 2

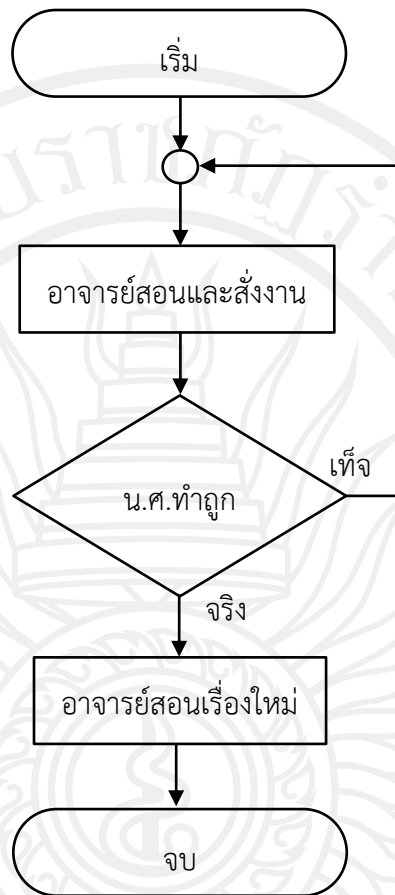
ทำซ้ำจนกระทั่งเงื่อนไขเป็นจริงหรือเท็จ
กิจกรรมที่ 1
จบการทำซ้ำ

ตัวอย่างที่ 2.9 การเขียนผังงานการสอนของอาจารย์แบบที่ 1
อัลกอริทึม

1. อาจารย์สอนและสั่งงาน
2. ทำซ้ำจนกระทั่งเงื่อนไขเป็นจริง
 นักศึกษาส่วนใหญ่ทำงานถูก
3. อาจารย์สอนเรื่องใหม่

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ผังงาน



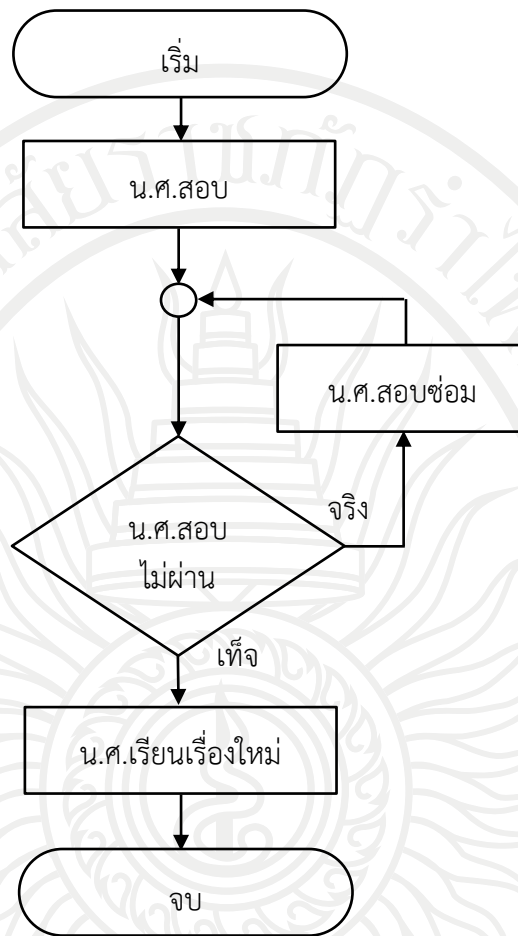
ในกรณีรูปแบบผังงานแบบทำซ้ำนี้ มีกระบวนการก่อนอย่างน้อย 1 กระบวนการ แล้วค่อยเข้าสู่เงื่อนไขของการตัดสินใจทำซ้ำ ถ้านักเรียนยังทำไม่ถูก ให้อนกลับไปทำที่อาจารย์สอนและสั่งงานใหม่อีกครั้งและให้เป็นแบบนี้ไปเรื่อย ๆ จนกว่านักเรียนทำงานถูก อาจารย์จึงสอนเรื่องใหม่และจบโปรแกรม

ตัวอย่างที่ 2.10 การเขียนผังงานการสอนของอาจารย์แบบที่ 2
อัลกอริทึม

1. นักศึกษาสอบ
2. ทำซ้ำจนกระทั่งเงื่อนไขเป็นจริง
นักศึกษาสอบไม่ผ่าน
3. นักศึกษาสอบซ่อม
4. นักศึกษาเรียนเรื่องใหม่

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ผังงาน



ในกรณีรูปแบบผังงานแบบทำซ้ำนี้ นักศึกษาทำการสอบก่อน จากนั้นจึงผ่านเข้าสู่เงื่อนไขของการตัดสินใจทำซ้ำว่าเป็นจริงหรือเท็จ ถ้าเป็นจริง นักศึกษาต้องผ่านกระบวนการก่อนอย่างน้อย 1 กระบวนการ นั่นคือนักศึกษาสอบซ่อม แล้วค่อยกลับไปเข้าสู่เงื่อนไขของการตัดสินใจทำซ้ำอีกครั้ง จนกว่าค่านักศึกษาสอบไม่ผ่านนั้นเป็นเท็จ จึงเข้าสู่กระบวนการถัดไปคือ น.ศ.เรียนเรื่องใหม่ และจบในลำดับถัดไป

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

สรุป

ในการเขียนโปรแกรมคอมพิวเตอร์อัลกอริทึมเป็นสิ่งสำคัญ เพราะเป็นกระบวนการแก้ปัญหาอย่างมีลำดับ จากนั้นนำมาบรรยายเป็นผังงานที่เปรียบเสมือนแผนที่ ที่ทำให้เราไม่หลงทางเวลาเขียนโปรแกรม โดยการเขียนผังงานแบ่งออกเป็น 3 ประเภทใหญ่ ๆ คือ ผังงานแบบตามลำดับที่เป็นการควบคุมการทำงานทีละคำสั่งจากบนลงล่าง ผังงานแบบเลือกเส้นทาง ที่เป็นการตัดสินใจโดยมีเงื่อนไขเป็นตัวกำหนด และเมื่อเลือกเส้นทางแล้วต้องมาพบกันและทำงานในขั้นต่อไปร่วมกันอีกครั้ง และ 3 ผังงานแบบทำซ้ำที่มีการทำงานแบบทำซ้ำคำสั่งโดยให้กลับมาทำงานขั้นตอนเก่าตามเงื่อนไขที่กำหนด ซึ่งผังงานทั้ง 3 ประเภท มีสัญลักษณ์ภาพในการเขียนที่กำหนดตามมาตรฐานเอเอ็นไอเอสไอและไอเอสไอ เพื่อให้ผู้ที่สนใจอื่น ๆ สามารถเข้าใจความคิดของผู้เขียนโปรแกรมได้

แบบฝึกหัดบทที่ 2

1. สัญลักษณ์การเขียนผังงานแสดง ข้อมูลนำเข้า/ส่งออก เป็นอย่างไร
2. อัลกอริทึม คืออะไร
3. จงเขียนผังงานแสดงการทอผ้าเช็ดตัว
4. จงเขียนผังงานแสดงการต้มมาล่า
5. การเขียนผังงานแบบเลือกเส้นทางมีทั้งหมดกี่แบบ อะไรบ้าง พร้อมทั้งยกตัวอย่าง
6. จงเขียนผังงานการรับข้อมูลจำนวนมะละกอที่ซื้อ เพื่อคำนวณหาราคามะละกอที่ต้องจ่ายเงิน โดยมีเงื่อนไขดังนี้
 ถ้าซื้อมะละกอมากกว่า 5 กิโลกรัมขึ้นไป คิดราคากิโลกรัมละ 40 บาท
 ถ้าซื้อมะละกอน้อยกว่าหรือเท่ากับ 4 กิโลกรัม คิดราคากิโลกรัมละ 30 บาท
7. จงเขียนผังงานเปรียบเทียบค่าข้อมูลที่เก็บอยู่ในตัวแปร A โดยมีเงื่อนไขดังนี้
 ถ้า $A > 0$ ให้พิมพ์คำว่า “A มากกว่า 0”
 ถ้า $A < 0$ ให้พิมพ์คำว่า “A น้อยกว่า 0”
 ถ้า $A = 0$ ให้พิมพ์คำว่า “A เท่ากับ 0”
8. จงเขียนผังงานตามอัลกอริทึมเพื่อประมวลผลต่อไปนี้
 1. รับค่า A และ B
 2. ประมวลผลตามสูตร $C = A + 50$
 3. เปรียบเทียบ B กับ C
 - 3.1 ถ้า $B \geq C$ พิมพ์ “ผ่าน”
 - 3.2 ถ้า $B < C$ พิมพ์ “ไม่ผ่าน”
 4. จบการทำงาน
9. การเขียนผังงานแบบทำซ้ำมีทั้งหมดกี่แบบ อะไรบ้าง
10. จงยกตัวอย่างอัลกอริทึม และผังงานแบบทำซ้ำ มา 2 เรื่อง

เอกสารอ้างอิง

- วิโรจน์ ชัยมูลและสุพรรณษา ยวงทอง, (2552). ความรู้เบื้องต้นเกี่ยวกับคอมพิวเตอร์และเทคโนโลยีสารสนเทศ. กรุงเทพมหานคร: โปรวิชั่น.
- สุรัตน์ โคอินทรางกูร, (ม.ป.ป.). การเขียน Flowchart เบื้องต้น. [On-line] Available : <http://www.bus.tu.ac.th/usr/surat/mis201/homepage/cmat/sheet/flowchar.doc>. [2556, ตุลาคม 27].
- โสภาส เอี่ยมสิริวงศ์, (2547). วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ. กรุงเทพมหานคร: ซีเอ็ดดูเคชั่น.

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

แผนบริหารการสอนประจำบทที่ 3

หัวข้อเนื้อหาบทที่ 3

1. การใช้แสง
2. การใช้สีขาว - ดำในการออกแบบ
3. การมีพื้นที่ว่าง
4. การวางตัวหนังสือบนรูปภาพ
5. การเพิ่มหรือลด ความเด่นของตัวหนังสือ
6. การเลือกใช้ชุดแบบอักษรให้เหมาะสม
7. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 3 แล้วนักศึกษาสามารถ

1. รู้จักและสามารถใช้แสงในการออกแบบหน้าจอได้
2. รู้จักและสามารถใช้สีขาว - ดำในการออกแบบหน้าจอได้
3. รู้จักและสามารถใช้พื้นที่ว่างในการออกแบบหน้าจอได้
4. รู้จักและสามารถวางตัวหนังสือบนรูปภาพได้
5. รู้จักและสามารถเพิ่มหรือลด ความเด่นของตัวหนังสือได้
6. รู้จักและสามารถเลือกใช้ชุดแบบอักษรให้เหมาะสมได้
7. สรุป

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายเนื้อหา ข้อตกลงในการเรียนการสอน
2. ให้นักศึกษาได้เรียนรู้ผ่านการบรรยายที่จอโปรเจคเตอร์
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 3

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
4. แบบฝึกหัดบทที่ 3

การวัดผลและประเมินผล

1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากกิจกรรมร่วม
3. สังเกตจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด

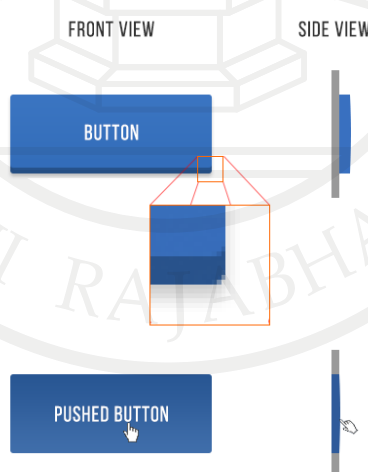
บทที่ 3

การออกแบบหน้าจอ

จากการศึกษาที่ผ่านมาเมื่อได้เรียนรู้วิธีการลงโปรแกรมและได้เรียนรู้วิธีการเขียนผังงานแล้ว สิ่งถัดไปที่ควรเรียนรู้คือ การออกแบบหน้าจอของโปรแกรม เนื่องจากการพัฒนาโปรแกรม สิ่งที่สำคัญที่สุดก็คือการสร้างส่วนที่ติดต่อกับผู้ใช้ในแบบกราฟฟิก (GUI:Graphic user interface) ซึ่ง วิศวกรเว็บ นั้นมีเครื่องมือจำนวนมากสำหรับใช้งาน แต่การเลือกใช้หรือเลือกวางแบบใดให้เกิดความเหมาะสมนั้นเป็นปัญหาใหญ่ของโปรแกรมเมอร์ นักพัฒนา หรือนักศึกษาที่ไม่ได้เรียนการออกแบบมา เพราะเวลาพัฒนาโปรแกรม, แอปพลิเคชันหรือเว็บไซต์ หลายคนขาดความสามารถในด้านการออกแบบ ไม่ว่าจะลองออกแบบอย่างไรก็เลือกสีที่ไม่โดนใจ เลือกชุดแบบอักษรได้ไม่สวย เป็นต้น แต่มีหลักการของเอียค ดี เคนเนดี้ (Erik D. Kennedy) นักวิศวกรที่ต้องทำงานทางด้านสร้างส่วนติดต่อกับผู้ใช้ (User interface) ได้เสนอวิธีออกแบบให้สวยงาม เหมาะกับบุคคลที่ไม่ได้ทำงานด้านออกแบบอย่างวิศวกร ด้วยหลักการง่าย ๆ ดังต่อไปนี้

การใช้แสง

เนื่องจากมนุษย์ชินกับการที่แสงส่องมาจากท้องฟ้าหรือด้านบน หากแสงมาจากด้านล่างจะทำให้งานออกแบบดูแปลกตาไปทันที ตลอดจนเมื่อมีแสงส่องลงมาย่อมต้องเกิด “เงา” อยู่ด้านล่างด้วย เพราะฉะนั้นด้านบนของสิ่งที่โดนแสงจะสว่างและด้านล่างจะมีมืดกว่าเสมอ การออกแบบเพื่อสร้างส่วนติดต่อกับผู้ใช้ด้วยหลักการของเคนเนดี้จึงเป็นไปตามเรื่องของแสง ปัจจุบันถ้าสังเกตในเว็บไซต์หรือแอปพลิเคชันต่าง ๆ ที่มีการออกแบบให้ผู้ใช้ต้องทำงานกับโปรแกรมที่นั้นก็ออกแบบพัฒนาขึ้น และได้ผ่านกระบวนการออกแบบจากดีไซน์เนอร์แล้ว จะพบว่า ปุ่ม มีการใส่ “เงา” เพื่อให้รู้สึกว่ามีปุ่มเป็น 3 มิติ ดูออกมาดังภาพที่ 3.1



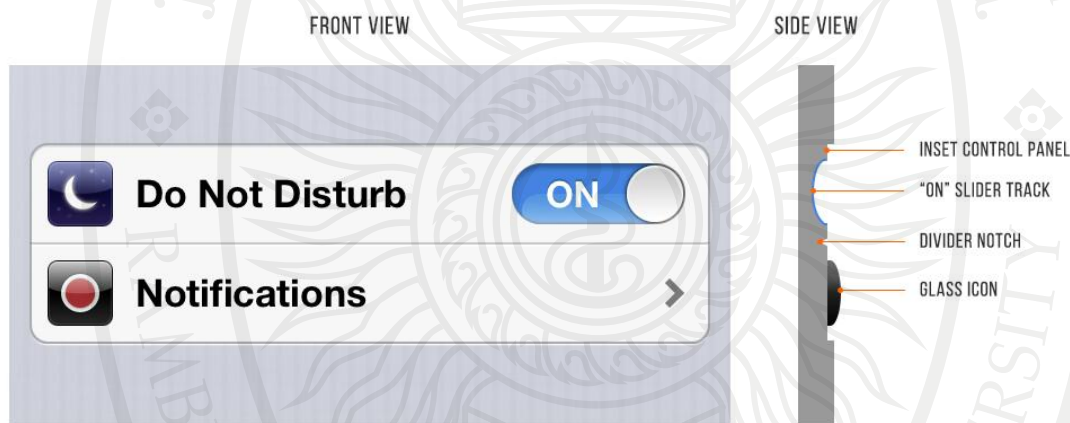
ภาพที่ 3.1 เทคนิคการใช้แสงในการออกแบบปุ่มบนเว็บไซต์หรือแอปพลิเคชัน
ที่มา (Erik D. Kennedy. n.d.)

จากภาพที่ 3.1 ปุ่มด้านบน แม้ดูเรียบง่าย แต่ก็มีการใช้รายละเอียดเรื่องแสงเข้ามาเกี่ยวข้อง ดังนี้

1. ปุ่มตอนที่ยังไม่กด (ภาพบน) มีมุมใต้ปุ่มเป็นสีเข้ม เนื่องจากตรงจุดนั้นแสงส่องไม่ถึง
2. ปุ่มตอนที่ยังไม่กด บนผิวปุ่มด้านบนเป็นสีน้ำเงินสว่างกว่าผิวปุ่มด้านล่าง เกิดจากผิวปุ่มด้านบนเอียงเล็กน้อยเพื่อรับแสง ส่วนผิวปุ่มด้านล่างเอียงตรงข้ามกับทิศทางแสง
3. ปุ่มตอนที่ยังไม่กด มีเงาออกมาเล็กน้อย (ในภาพจุดที่ซูม)
4. ปุ่มตอนที่กดแล้ว (ภาพล่าง) พบว่าสีของปุ่มเข้มกว่าตอนแรก เนื่องจากพอกดปุ่มไปแล้วแสงจะส่องไม่ถึง เช่นเดียวกับการใช้งานจริง

ทั้งนี้ถ้าใส่ใจในเรื่องของแสงเพียงแค่ปุ่มกดธรรมดาปุ่มเดียว ยังประกอบไปด้วยการใช้รายละเอียดเรื่องแสงถึง 4 อย่าง ด้วยกัน

ตัวอย่าง ลองสังเกตการใช้แสงในการออกแบบของระบบไอโอเอสดังภาพที่ 3.2 สังเกตว่าในหน้านี้มีการใช้รายละเอียดเรื่องแสงมาประกอบการออกแบบมากมาย



ภาพที่ 3.2 การใช้แสงในการออกแบบรอยต่อที่มา (Erik D. Kennedy. n.d.)

1. กรอบขาวของช่องมีเงาเล็ก ๆ อยู่ด้านบน เกิดจากการที่กรอบนี้ยุบลงไปในตัว
2. ตัวเลื่อน “ON” ที่ยุบลงไปเหมือนกัน ก็เกิดเงาเล็ก ๆ ด้านบนเช่นกัน
3. ตัวเลื่อน “ON” ที่ยุบลง ทำให้ด้านล่างของตัวเลื่อนรับแสงเยอะกว่าด้านบน เกิดเป็นสีฟ้าด้านล่างและสีน้ำเงินด้านบน
4. ไอคอนนูนออกมาเล็กน้อย ทำให้ด้านบนของไอคอนรับแสงมากกว่าปกติ เกิดเป็นแสงสีขาวด้านบนไอคอนทั้ง 2 อัน
5. เส้นคั่นระหว่าง “Do Not Disturb” กับ “Notifications” เกิดจากจุดที่ยุบลงไปทำให้เกิดเงาสีเข้มขึ้นมา

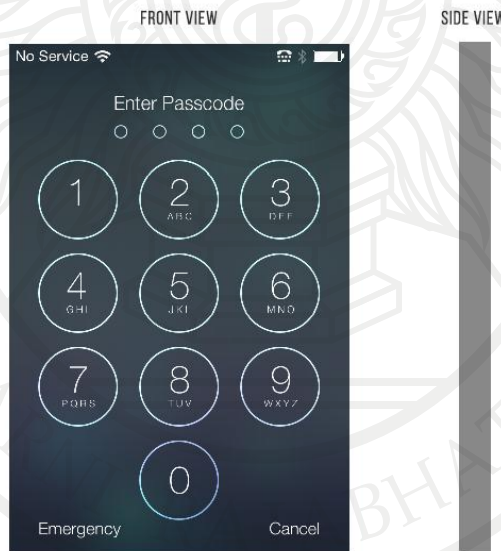
ดังนั้นเมื่อรู้ว่ามีสิ่งที่ “นูนขึ้น” กับ “นูนลง” ในการออกแบบจะสังเกตเห็นว่ามีนูนอยู่เกือบทุกที่เกือบทุกแอปพลิเคชันหรือเกือบทุกเว็บไซต์ ซึ่งอาจสรุปได้ว่า การออกแบบเพื่อสร้างส่วนติดต่อกับผู้ใช้ด้วย ที่มีกนูนลง (Inset) จะอยู่ที่

1. ช่องกรอกในฟอร์มต่าง ๆ
2. ปุ่มตอนที่กดแล้ว
3. รางของตัวเลื่อน (แบบใต้ปุ่ม “ON” ในไอโอเอสด้านบน)
4. ปุ่มบัตอนตอนที่ยังไม่เลือก

และที่มีกนูนขึ้น (Outset):

1. ปุ่มตอนที่ยังไม่กด
2. ตัวเลื่อน (แบบปุ่ม “ON” ในไอโอเอสด้านบน)
3. ปุ่มเปิด
4. ปุ่มบัตอนตอนที่เลือกแล้ว

นอกจากนี้ ยังมีการออกแบบแบบใหม่ที่เรียกกันว่าแฟลทดีไซน์ (Flat Design) ที่ไม่มีการนูนหรือการใช้เงา ดังภาพที่ 3.3 โดยมีแค่การใช้เส้นกับรูปทรงที่ใช้สีเดียวกันทั้งก่อน เพื่อให้งานดูเรียบ ซึ่งอาจดูสะอาดและความเรียบง่าย แต่การขาดเงาไป ทำให้รู้สึกไม่เป็นธรรมชาติ เพราะฉะนั้นในอนาคตการออกแบบน่าจะกำลังขยับไปทาง กิ่งเรียบที่ใช้แสงเงามาประกอบด้วยดังภาพที่ 3.4

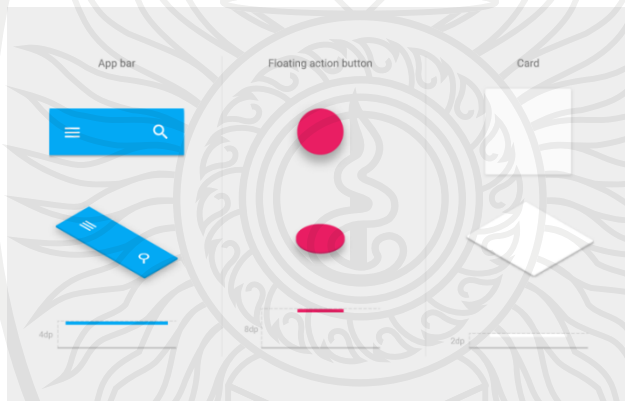


ภาพที่ 3.3 การออกแบบปุ่มการเชื่อมต่อกับผู้ใช้แบบเรียบง่าย
ที่มา (Erik D. Kennedy. n.d.)

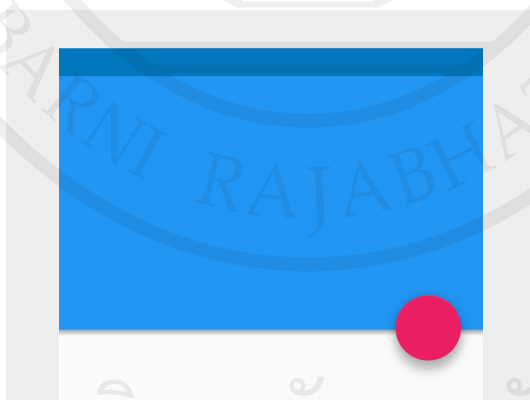


ภาพที่ 3.4 โปรแกรมที่ถูกออกแบบแบบกึ่งเรียบ
ที่มา (Erik D. Kennedy. n.d.)

อีกทั้งกูเกิ้ลได้มีการเปิดตัวการออกแบบวัสดุ ซึ่งเป็นหลักที่ใช้ออกแบบผลิตภัณฑ์ในเครื่องกูเกิ้ลทั้งหมด โดยเน้นการแสดงความรู้สึกของสิ่งต่าง ๆ ด้วยแสงเงา เลียนแบบเงาของสิ่งของดังภาพที่ 3.5 และภาพที่ 3.6



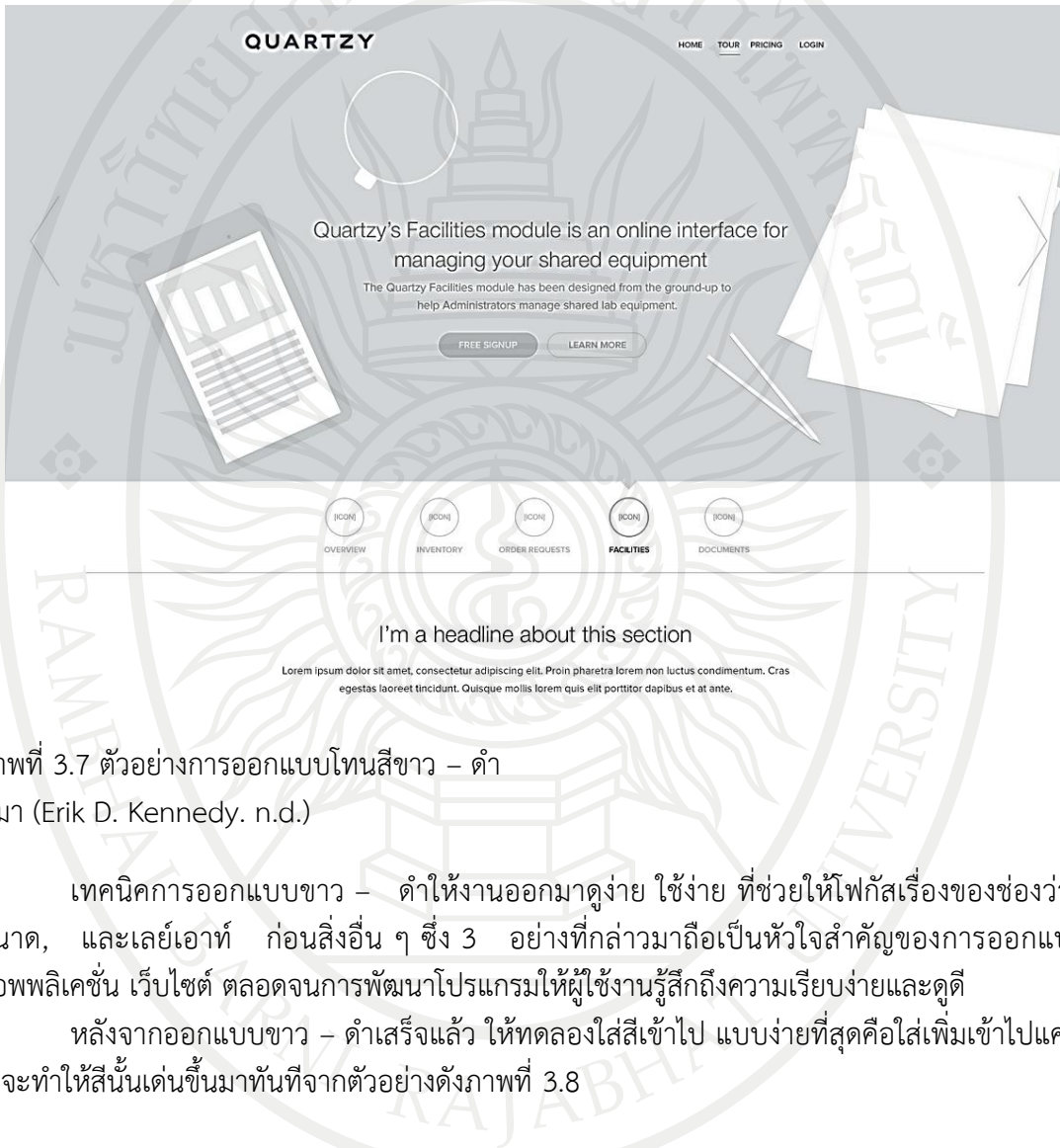
ภาพที่ 3.5 การออกแบบวัสดุที่ใช้แสงเงาเพิ่มความรู้สึก
ที่มา (Erik D. Kennedy. n.d.)



ภาพที่ 3.6 การใช้แสงเงาทำให้เกิดความรู้สึก ความนูน
ที่มา (Erik D. Kennedy. n.d.)

การใช้สี ขาว - ดำ ในการออกแบบ

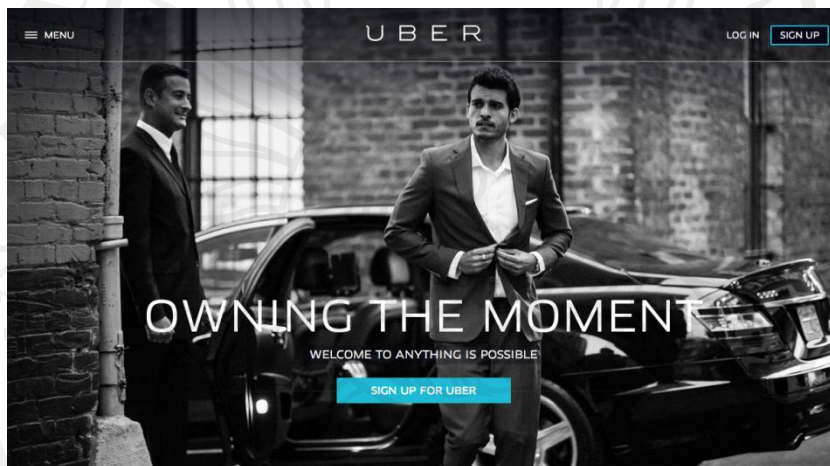
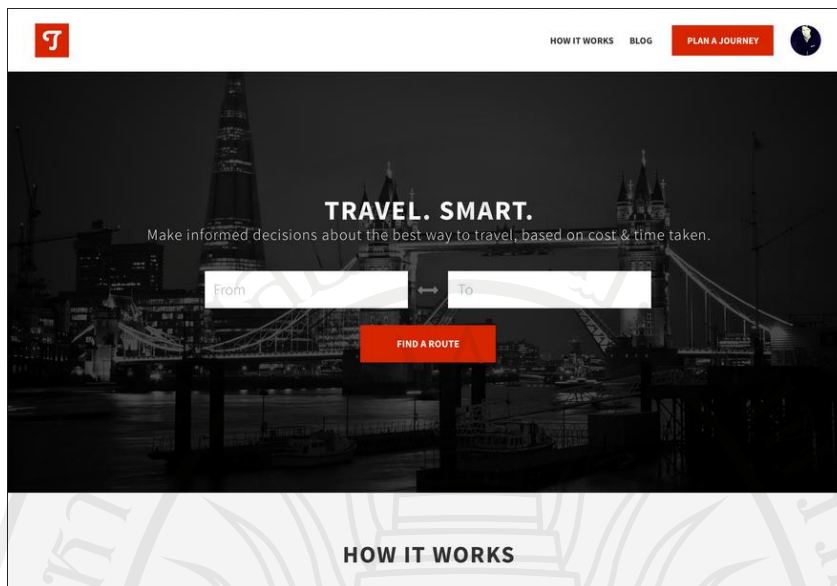
โดยมีแนวคิดคือ การออกแบบเป็นสีขาว - ดำก่อน จะช่วยให้โฟกัสไปในเรื่องของการจัดวางวัตถุหรือสิ่งต่าง ๆ ให้เข้าที่ การจัดช่องว่าง และวางเลย์เอาต์สิ่งต่าง ๆ และคิดวิธีให้งานออกมาดูสวยและใช้งานง่าย โดยไม่ต้องพึ่งการใช้สีเลยแม้แต่หน่อย แล้วหลังจากนั้นอาจค่อยใส่สีเข้าไปทีละนิด ซึ่งการใส่สีต้องมีจุดประสงค์ในการใส่ด้วยทุกครั้ง ดังตัวอย่างภาพที่ 3.7



ภาพที่ 3.7 ตัวอย่างการออกแบบโทนสีขาว - ดำ
ที่มา (Erik D. Kennedy. n.d.)

เทคนิคการออกแบบขาว - ดำให้งานออกมาดูง่าย ใช้งาน ที่ช่วยให้โฟกัสเรื่องของช่องว่าง, ขนาด, และเลย์เอาต์ ก่อนสิ่งอื่น ๆ ซึ่ง 3 อย่างที่กล่าวมาถือเป็นหัวใจสำคัญของการออกแบบแอปพลิเคชัน เว็บไซต์ ตลอดจนการพัฒนาโปรแกรมให้ผู้ใช้งานรู้สึกถึงความเรียบง่ายและดูดี

หลังจากออกแบบขาว - ดำเสร็จแล้ว ให้ทดลองใส่สีเข้าไป แบบง่ายที่สุดคือใส่เพิ่มเข้าไปแค่ 1 สี จะทำให้สีนั้นเด่นขึ้นมาทันทีจากตัวอย่างดังภาพที่ 3.8

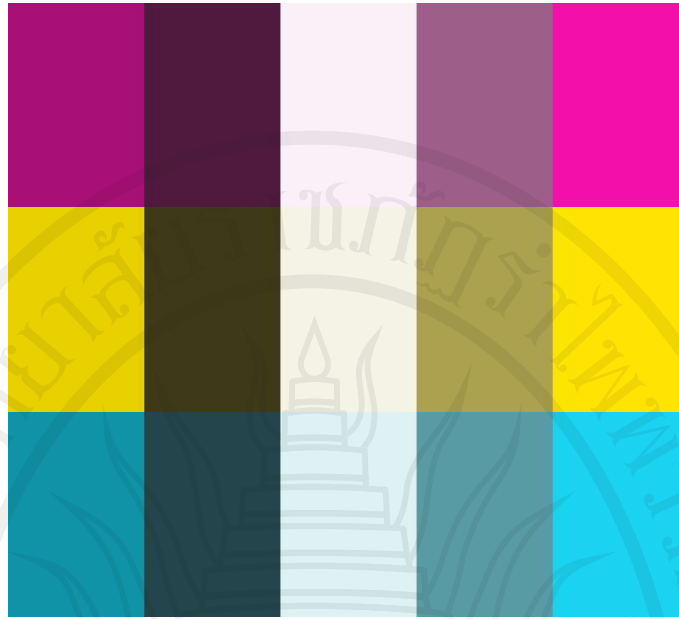


ภาพที่ 3.8 ตัวอย่างการออกแบบโทนสีขาว – ดำ ที่เพิ่มสีเข้าไปอีก 1 สี
ที่มา (Erik D. Kennedy. n.d.)

ซึ่งนักพัฒนาอาจพัฒนาไปอีกขั้น โดยการใส่สี 2 สี, 3 สี, 4 สี ไปเรื่อย ๆ ก็ได้ แต่ควรใช้สีที่มาจากเนื้อสี (Hue) เดียวกัน เช่น ถ้าใส่สีแดง อาจจะใส่แดงธรรมชาติ แดงเข้ม แดงอ่อน เป็นต้น

ทั้งนี้สีอาร์จีบี (RGB) คือการสร้างสีหนึ่งขึ้นมาจากการรวมของสี 3 สี คือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) แต่สีเอชเอสบี (HSB) สร้างสีขึ้นมาจากการรวมกันของ เนื้อสี ความเข้มข้น (Saturation) และความสว่าง (Brightness) เพราะฉะนั้นถ้าใช้สีแบบเฮชเอสบี จะทำให้สามารถตั้งเนื้อสีคงเดิม (เช่น สีแดง) แล้วเปลี่ยนเฉพาะความเข้มข้นกับความสว่างเพื่อให้ได้สีใหม่ที่เป็นโทนเดียวกัน (เช่น สีแดงเข้ม สีแดงอ่อน) ตัวอย่างเป็นการปรับสีแบบ เนื้อสีคงที่แล้วเปลี่ยนแค่ความเข้มข้นกับความสว่าง ดังภาพที่ 3.9

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 3.9 ตัวอย่างการเปลี่ยนความเข้มข้นและความสว่างของโทนสีโดยที่เนื้อสีคงที่
ที่มา (Smashing magazine. 2010)

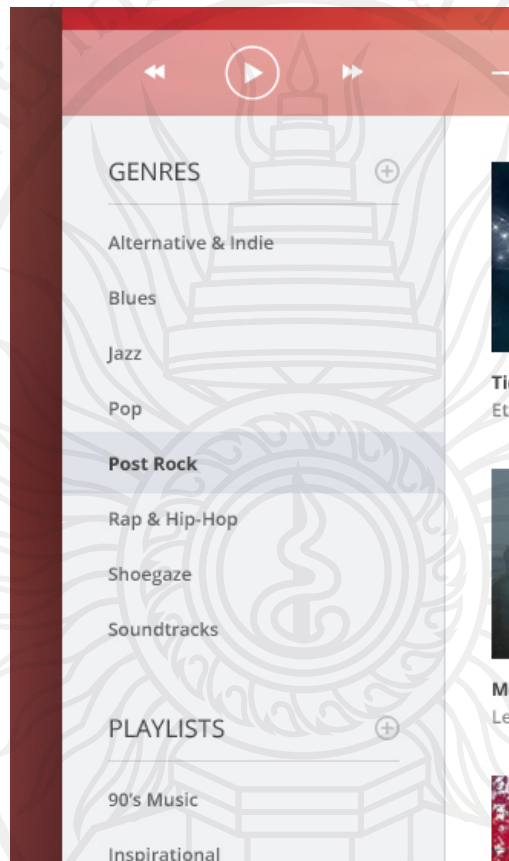
เทคนิคการเลือกใช้สีในงานออกแบบที่มาจากเนื้อสีหลักไม่เกิน 1-2 เนื้อสี ทำให้สามารถแบ่ง
ส่วนของการออกแบบที่ต้องการให้เด่นได้ง่าย ๆ โดยที่ไม่ทำให้งานออกแบบโดยรวมดูเลอะเทอะ ดังภาพที่
3.10



ภาพที่ 3.10 ตัวอย่างการออกแบบนาฬิกาจับเวลา
ที่มา (Kerem. n.d.)

การมีพื้นที่ว่าง

การมีพื้นที่ว่าง (Whitespace) หมายถึงพื้นที่โล่ง ๆ รอบ ๆ ทุกจุดในงานออกแบบ เน้นความง่าย สบายตามากกว่าพยายามอัดทุกอย่างไว้ในหน้าเดียวกัน โดยให้พยายามใส่พื้นที่ว่างในทุก ๆ อย่าง ไม่ว่าจะเป็นตัวหนังสือแต่ละบรรทัด, ระยะห่างระหว่างส่วนประกอบสำคัญต่าง ๆ บนโปรแกรมและแอปพลิเคชัน ฯลฯ จะทำให้งานออกมาดูดีขึ้นในทันทีดังภาพที่ 3.11



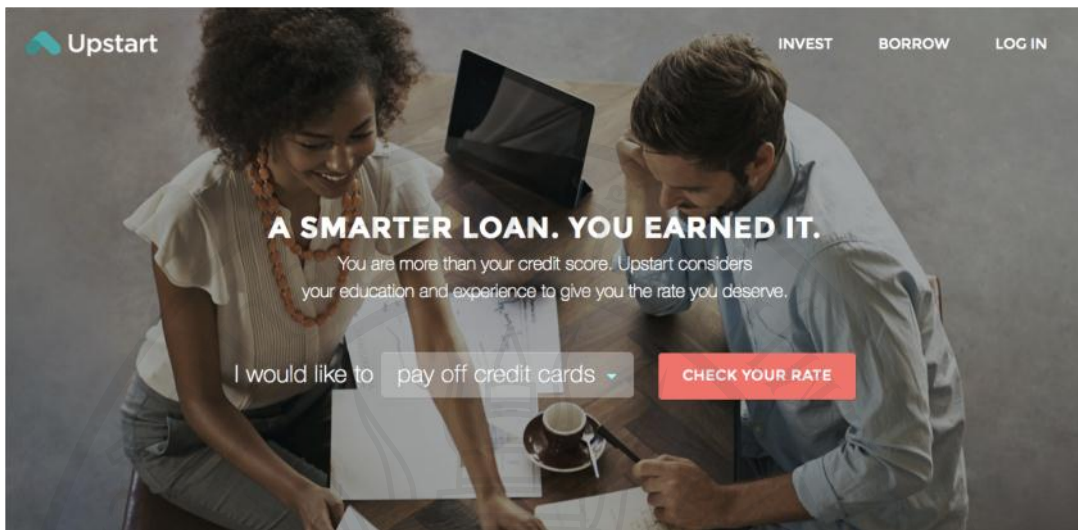
ภาพที่ 3.11 ตัวอย่างโปรแกรมเล่นเพลง
ที่มา (Piotr Kwiatkowski. n.d.)

จากภาพที่ 3.11 สังเกตว่าระยะห่างแนวตั้งระหว่างแต่ละเมนู มีความสูงประมาณ 2 เท่าของขนาดตัวอักษร หรือระยะห่างแนวตั้งระหว่างหัวข้อกับเส้นใต้หัวข้อที่สูงและด้วยการใช้พื้นที่ว่างเยอะทำให้งานออกมาดูไม่แน่น อ่านสบาย ๆ และดูสวยงาม

การวางตัวหนังสือบนรูปภาพ

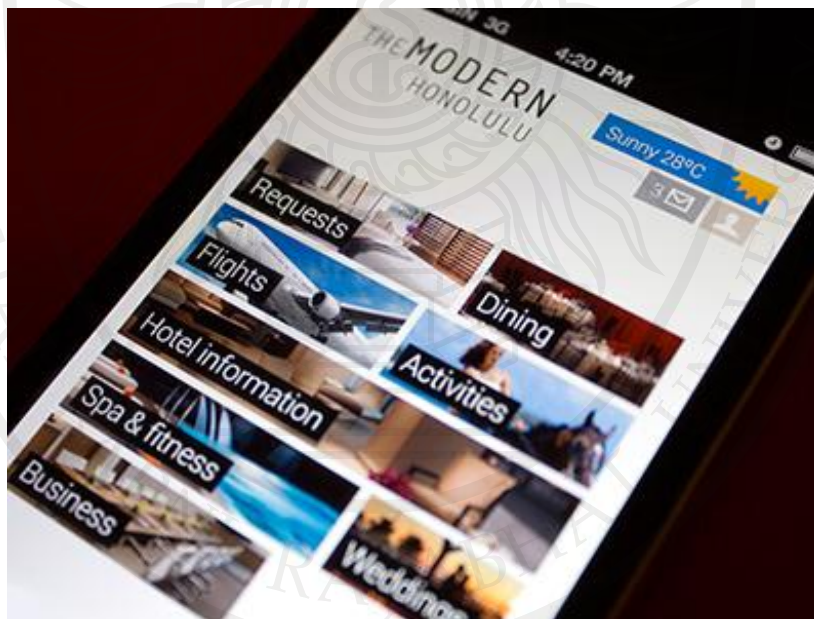
การวางข้อความบนรูปภาพ นั้นยากที่จะทำให้ข้อความนั้นเด่นชัด แต่มีเทคนิคง่าย ๆ ต่อไปนี้คือ

1. ใช้สีที่บัพพื้นที่หลัง (Overlay) ในกรณีที่ รูปสว่างเกินไปหรือมีจุดที่สีตัดกันเยอะมากจนวางข้อความไม่ได้ หรือวางแล้วไม่สวย ให้ใช้สีกึ่งโปร่งใสที่เป็นสีเข้มมาทับรูปจะทำให้ข้อความดูขาวและชัดขึ้น



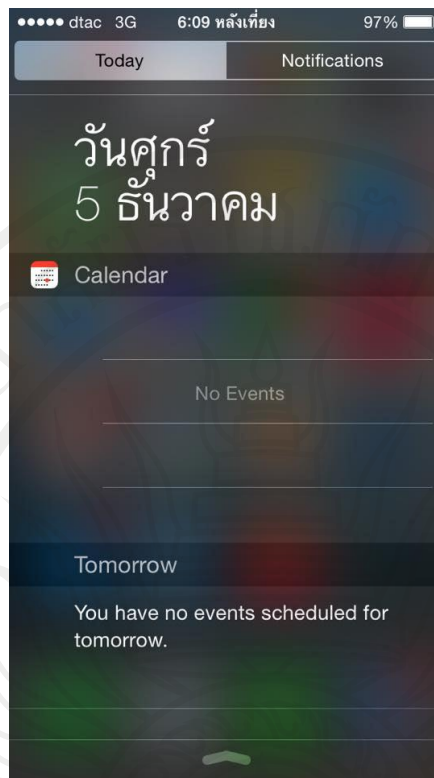
ภาพที่ 3.12 ตัวอย่างการใช้สีดำโปร่งใสประมาณ 30% เพื่อให้ตัวหนังสืออ่านง่ายขึ้น
ที่มา (Upstart. 2012.)

2. ใส่ตัวอักษรไว้ในกล่อง เทคนิคนี้ง่ายยิ่งเทคนิคแรก โดยเฉพาะใส่ตัวหนังสือขาวในกล่องพื้น
หลังสีดำ หรือสีดำแบบโปร่งใสก็ได้ หรือพื้นหลังกล่องจะเป็นสีอื่นที่ไม่ใช่สีดำก็ได้



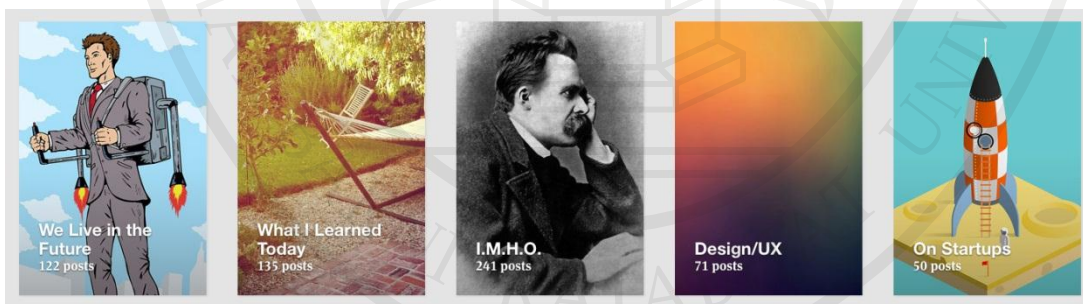
ภาพที่ 3.13 ตัวอย่างงานออกแบบโปรแกรมแบบใส่ตัวอักษรไว้ในกล่อง
ที่มา (Miguel oliva márquez. n.d.)

3. เบลอรูปพื้นหลัง เทคนิคนี้มาพร้อมกับไอโอเอส ที่ใช้พื้นหลังเบลอ ๆ กับข้อความซึ่งทำให้เรา
เห็นว่าการวางตัวหนังสือบนรูปที่เบลอนั้นอ่านง่ายสุด ๆ



ภาพที่ 3.14 ตัวอย่างงานเบลอรูปพื้นหลัง

4. สีผิวบาง ๆ ด้านล่าง เทคนิคนี้คล้ายกับเทคนิคที่ 1 ที่เอาสีดำไปรงใส่ทับรูป แต่เทคนิคนี้ใช้ทับเฉพาะส่วนล่างที่มีตัวหนังสือ แล้วค่อย ๆ ไล่สีออกจนมองไม่เห็นและการไล่สีเข้มไว้ด้านล่างจะทำให้ดูแล้วไม่รู้สึกละเอียด เพราะไม่ขัดกับแสงธรรมชาติ



ภาพที่ 3.15 ตัวอย่างการออกแบบให้พื้นผิวด้านล่างเลื่อนรางที่มา (Medium. n.d.)

การเพิ่มหรือลด ความเด่นของตัวหนังสือ

การทำให้ “เด่นขึ้น” คือ เป็นการทำให้ตัวหนังสือจุดนั้นมองเห็นได้ชัดเจนกว่าจุดอื่น ไม่ว่าจะเป็นการทำตัวหนังสือใหญ่, ชุดแบบอักษรหนา, ตัวพิมพ์ใหญ่ ฯลฯ ส่วนการทำให้ “เด่นน้อยลง” เป็นการทำให้ตัวหนังสือจุดนั้นมองเห็นได้ยากกว่าจุดสำคัญอื่น ไม่ว่าจะเป็นการทำให้ตัวหนังสือเล็ก, ชุดแบบ

อักษรบาง ฯลฯ วิธีที่ทำให้ตัวหนังสือดูเด่นขึ้นหรือเด่นน้อย จากการเลือกน้ำหนักให้ตัวหนังสืออย่างเหมาะสมนั้นเป็นสิ่งที่ยากพอ ๆ กับการวางตัวหนังสือลงในภาพ โดยมีขอแนะนำดังนี้

1. เปลี่ยนขนาดตัวหนังสือ
2. เปลี่ยนสีตัวหนังสือ
3. ปรับตัวหนา-บาง
4. ใช้ตัวเอียง
5. เปลี่ยนระยะห่างตัวหนังสือ
6. เปลี่ยนระยะห่างของกล่อง
7. การใช้ตัวหนังสือมีหัว - ไม่มีหัว ดังภาพที่ 3.16



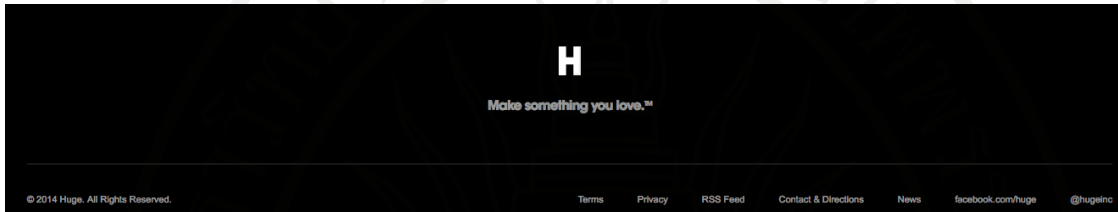
ภาพที่ 3.16 ตัวอย่างชุดแบบอักษรไทย แบบมีหัวและไม่มีหัว

ตัวอย่าง เพิ่มและลด ความเด่นของตัวหนังสือ จากเว็บไซต์เฮชยูจีอี (Huge) ซึ่งเป็นหน่วยงานในต่างประเทศที่ดังมาก โดยสังเกตตรงคำว่า “Material Design” ในเว็บไซต์มีการใช้เทคนิคที่ทำให้ตัวหนังสือดู “เด่นขึ้น” ไม่ว่าจะเป็น การใช้ชุดแบบอักษรหนา , ตัวหนังสือใหญ่, การมีพื้นที่ว่างเยอะ ๆ , ตัวหนังสือสีดำตัดกับสีขาวมาก ๆ เมื่อดูเว็บไซต์นี้แล้วจะเห็นคำว่า “Material Design” ก่อนแน่นอน ดังภาพที่ 3.17



ภาพที่ 3.17 ตัวอย่างการเพิ่มความเด่นของตัวหนังสือที่มา (Huge. n.d.)

ในหน้าเว็บไซต์เดียวกัน ถ้าเลื่อนลงมาด้านล่างจะพบเทคนิคการทำให้ “เด่นน้อยลง” สังเกตตรงเมนูด้านล่าง จะเห็นว่าใช้เทคนิคทำให้ตัวหนังสือ “เด่นน้อยลง” นั่นคือ ใช้ตัวหนังสือสีเทาบนพื้นหลังสีดำ ทำให้อ่านได้ยาก นอกจากนั้นยังใช้ตัวหนังสือบางและเล็กอีกด้วย ดังภาพที่ 3.18 ซึ่งหน้าเว็บไซต์ที่ดีต้องมีการบาลานซ์ทั้งส่วนที่ทำให้ “เด่นขึ้น” และ “เด่นน้อยลง” อย่างพอเหมาะ ซึ่งในบางสถานการณ์อาจใช้ทั้งเทคนิคที่ทำให้ “เด่นขึ้น” และ “เด่นน้อยลง” ในจุดเดียวกัน เพื่อให้เกิดบาลานซ์ ดังภาพที่ 3.19



ภาพที่ 3.18 ตัวอย่างการลดความเด่นของตัวหนังสือที่หมา (Huge. n.d.)



ภาพที่ 3.19 ตัวอย่างการออกแบบที่ใช้เทคนิคเด่นขึ้นและเด่นน้อยลงในจุดเดียวกันที่หมา (Bluhomes n.d.).

การเลือกใช้ชุดแบบอักษรให้เหมาะสม

ชุดแบบอักษรเป็นตัวกำหนดอารมณ์ของงานออกแบบที่สำคัญมากเลยครับ ชุดแบบอักษรแต่ละแบบให้ความรู้สึกแตกต่างกันไป ถ้าอยากทำงานออกแบบให้ดูร่วมสมัย, สะอาด จำเป็นต้องเลือกชุดแบบอักษรที่ให้สื่อไปในทางเดียวกันด้วย โดยสามารถโหลดชุดแบบอักษรไทยฟรีได้ดังนี้

1. Font.com เว็บไซต์นี้มีชุดแบบอักษรไทยแจกฟรี ตระกูล JS, DS, PLE และอื่น ๆ นอกจากนี้ยังมีบทความเกี่ยวกับชุดแบบอักษรที่น่าอ่าน และเว็บบอร์ดที่ให้คนชอบชุดแบบอักษรเข้าไปพูดคุยกัน

2. 13 ฟอนต์ไทย สำหรับนักออกแบบที่หาชุดแบบอักษรไทยสวย ๆ ที่เหมาะกับเว็บ โดยมีชุดแบบอักษรไทยฟรีถึง 13 ตัวด้วยกัน

3. อาร์เอสยู ฟอนท์ (RSU Fonts) ทางมหาวิทยาลัยรังสิตได้ทำชุดแบบอักษรฟรี 2 ตัว ตัวแรกเป็นชุดแบบอักษรแนวโมเดิร์นชื่อ อาร์เอสยู (RSU) กับอีกตัวเป็นชุดแบบอักษรลายมือ

สรุป

หลักการง่าย ๆ ของเคนเนดีสำหรับการออกแบบเพื่อสร้างส่วนที่ติดต่อกับผู้ใช้ คือ 1) การใช้แสงจากสภาพความเคยชินของมนุษย์ ที่เมื่อมีแสงแล้วต้องเกิดเงา และแสงควรมาจากด้านบน 2) การใช้สีขาว – ดำในการออกแบบก่อน เพื่อให้เกิดความสนใจไปที่การวางตำแหน่งขององค์ประกอบแล้วจึงค่อยเพิ่มสีที่ต้องการ 3) การเพิ่มพื้นที่ว่างให้มาก ๆ เพื่อให้ดูสะอาดและสบายตา 4) หลักการวางตัวหนังสือบนรูปภาพด้วยเทคนิคต่าง ๆ เช่น วางในกล่อง ทำข้างหลังให้เบลอ ฯลฯ 5) การเพิ่มและลดความเด่นของตัวหนังสือเพื่อมุ่งเน้นจุดสะดุดตาหรือจุดสนใจ 6) การเลือกใช้ชุดแบบอักษรให้เหมาะสมซึ่งทั้งหมดนี้ยังมีอีกอย่างที่น่าสนใจคือ การมองงานของผู้อื่นให้มาก ๆ เพราะนั่นคือแบบที่เก๋ ย่อมต้องผ่านการเป็นนักออกแบบที่ไม่เก๋มาก่อนและการลองทำงานออกแบบตามคนอื่นเพื่อศึกษาก็เป็นวิธีที่ดีวิธีหนึ่งในการพัฒนาเช่นกัน

แบบฝึกหัดบทที่ 3

1. จงร่างแบบปุ่มกด ในแอปพลิเคชันบนมือถือ มา 2 แบบ
2. จงหาแอปพลิเคชันหรือเว็บไซต์ใดก็ได้ที่มีการใช้แสงตามหลักการของเคนเนดีมา 5 รายการ
3. จงร่างแบบหน้าจอเมนูควบคุมอะไรก็ได้ โดยใช้สีขาวดำในการออกแบบ
4. จงเลือกภาพมา 3 ภาพ แล้วใช้เทคนิคสีที่รูปพื้นหลังของเคนเนดีในการใส่ข้อความลงบนภาพ
5. จงเลือกภาพมา 3 ภาพ แล้วใช้เทคนิคใส่ตัวอักษรไว้ในกล่องของเคนเนดีเพื่อบรรยายภาพ
6. จงเลือกภาพมา 3 ภาพ แล้วใช้เทคนิคสีผิวบาง ๆ ด้ายล่างของเคนเนดีในการใส่ข้อความลงบนภาพ
7. จงยกตัวอย่างแบบอักษรไทยแบบมีหัวและไม่มีหัวมาอย่างละ 5 แบบ
8. จงร่างแบบหน้าจอแนะนำตัวเองโดยใช้เทคนิคของเคนเนดี
9. จงร่างแบบหน้าจอเมนูการจองตั๋วภาพยนตร์สำหรับบุคคลทั่วไปที่จะรับชมภาพยนตร์
10. จงศึกษาเรื่องการคิด ดัชนีมวลกาย (BMI: Body mass index) และออกแบบหน้าจอคำนวณค่าดัชนีมวลกาย

เอกสารอ้างอิง

- Bluhomes (n.d.). A better client experience. [On-line] Available : <https://www.bluhomes.com/our-vision>[2556, November 24].
- Erik D. Kennedy (n.d.). 7 Rules for Creating Gorgeous UI part 1. [On-line]. <https://medium.com/@erikdkennedy/7-rules-for-creating-gorgeous-ui-part-1-559d4e805cda#.8v06ghkou>. [2556, November 10].
- Erik D. Kennedy (n.d.). 7 Rules for Creating Gorgeous UI part 2. [On-line]. <https://medium.com/@erikdkennedy/7-rules-for-creating-gorgeous-ui-part-2-430de537ba96#.ypo0j45u8>. [2556, November 10].
- Huge (n.d.). A new home for Google’s universal design language, Material Design, that brings the new spec to life on any size screen. [On-line] Available : <http://www.hugeinc.com/case-study/material-design/>[2556, November 24].
- Kerem (n.d.). Kerem Suer designs systematic methods to help humans interact with machines. [On-line] Available : <http://kerem.co/>. [2556, November 11].
- Medium (n.d.). Floor Fade [On-line] Available : <https://medium.com/>[2556, November 24].
- Miguel Oliva Márquez (n.d.). Modern Honolulu [On-line] Available : <http://www.designil.com/miguelolivamarquez.com> [2556, November 24].
- Piotr Kwiatkowski (n.d.). Music UI – UI Design Concept [On-line] Available : <http://www.piotrkwiatkowski.co.uk/>[2556, November 11].
- Smashing Magazine (2010). Color Theory For Designers: Creating Your Own Color Palettes. [On-line] Available : <https://www.smashingmagazine.com/2010/02/color-theory-for-designer-part-3-creating-your-own-color-palettes/> [2556, November 11].
- Upstart (2012.). logo [On-line] Available : <https://www.upstart.com/>[2556, November 24].

แผนบริหารการสอนประจำบทที่ 4

หัวข้อเนื้อหาบทที่ 4

1. การสร้างโครงงานใหม่
2. คอนโทรลบัดทอน
3. คอนโทรลโอวีลเซพ
4. คอนโทรลฟิกเจอร์บ็อกซ์
5. คอนโทรลลาเบล
6. คอนโทรลเท็กซ์บ็อกซ์
7. คอนโทรลเช็คบ็อกซ์
8. คอนโทรลเรดิโอบัดทอน
9. คอนโทรลกรุปบ็อกซ์
10. คอนโทรลสโครลบาร์
11. คอนโทรลไทมเมอร์
12. การตั้งชื่อคอนโทรลและอ็อบเจ็กต์ตามคำแนะนำของไมโครซอฟท์
13. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 4 แล้วนักศึกษาสามารถ

1. รู้จักและสามารถสร้างโครงงานใหม่ได้
2. รู้จักและสามารถใช้คอนโทรลบัดทอนได้
3. รู้จักและสามารถใช้คอนโทรลโอวีลเซพได้
4. รู้จักและสามารถใช้คอนโทรลฟิกเจอร์บ็อกซ์ได้
5. รู้จักและสามารถใช้คอนโทรลลาเบลได้
6. รู้จักและสามารถใช้คอนโทรลเท็กซ์บ็อกซ์ได้
7. รู้จักและสามารถใช้คอนโทรลเช็คบ็อกซ์ได้
8. รู้จักและสามารถใช้คอนโทรลเรดิโอบัดทอนได้
9. รู้จักและสามารถใช้คอนโทรลกรุปบ็อกซ์ได้
10. รู้จักและสามารถใช้คอนโทรลสโครลบาร์ได้
11. รู้จักและสามารถใช้คอนโทรลไทมเมอร์ได้
12. รู้จักการตั้งชื่อคอนโทรลและอ็อบเจ็กต์ตามคำแนะนำของไมโครซอฟท์

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายเนื้อหา ข้อตกลงในการเรียนการสอน
2. ให้นักศึกษาได้เรียนรู้ผ่านการบรรยายที่จ้อปรเจคเตอร์
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 4

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. โปรแกรมคอมพิวเตอร์
4. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
5. แบบฝึกหัดบทที่ 4

การวัดผลและประเมินผล

1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากกิจกรรมร่วม
3. สังเกตจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

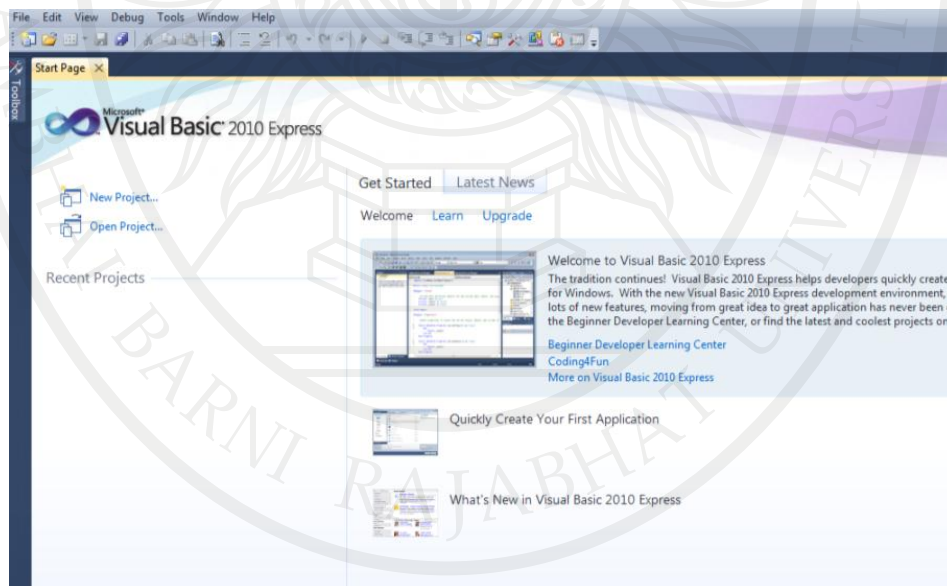
บทที่ 4

การใช้งานคอนโทรลเบื้องต้น

หลังจากได้ศึกษาการออกแบบหน้าจอด้วยหลักการของเคนเนดีแล้ว ในบทนี้จึงเป็นการนำความรู้จากการศึกษาการออกแบบมาใช้ในโปรแกรม ซึ่งอุปกรณ์ที่ใช้ในวิชวลเบสิกนี้เรียกว่าคอนโทรล ซึ่งมีมากมายหลายชนิด จึงหยิบยกคอนโทรลเบื้องต้นที่นิยมใช้บ่อย ๆ ในการทำงานเพื่อเขียนโปรแกรมควบคุมต่าง ๆ อันได้แก่คอนโทรลปุ่มกด, คอนโทรลโอเวิลเชพ, คอนโทรลฟลักเจอร์บ็อกซ์, คอนโทรลลาเบล, คอนโทรลเท็กซ์บ็อกซ์, คอนโทรลเช็คบ็อกซ์, คอนโทรลเรดิโอปุ่มกด, คอนโทรลกรุปบ็อกซ์, คอนโทรลสไลด์บาร์ และคอนโทรลไทมเมอร์ มาเป็นแนวทางเพื่อศึกษาและทำความเข้าใจ โดยในแต่ละปุ่มกดนั้นมีคุณสมบัติที่ใช้สำหรับการเขียนหรืออ้างถึงเพื่อใช้งานให้ปุ่มคอนโทรลเหล่านั้นเกิดการเปลี่ยนแปลง พร้อมทั้งมีตัวอย่างเพื่อนักศึกษาได้นำไปประยุกต์ใช้เพื่อพัฒนาโปรแกรมสำหรับวิศวกรรมต่อไป

การสร้างโครงการใหม่

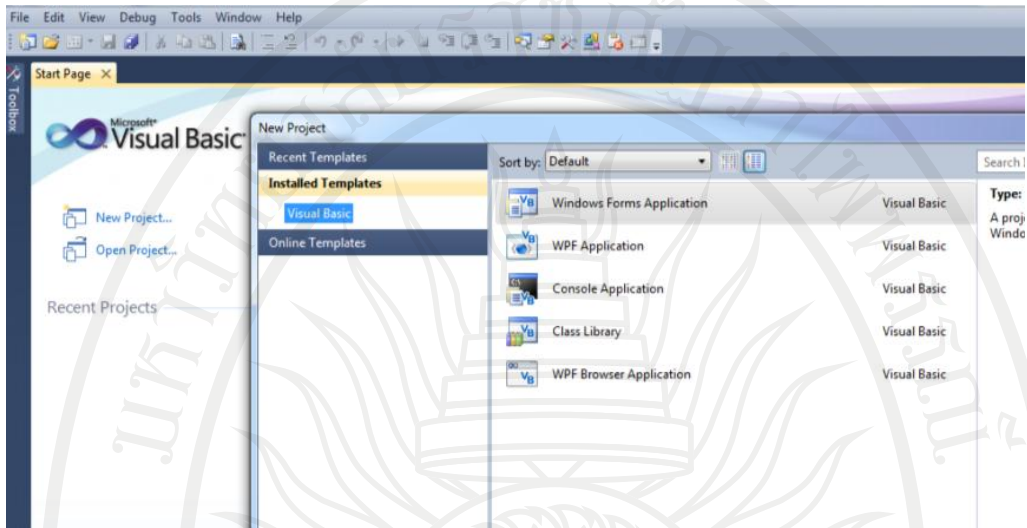
การเปิดโครงการขึ้นมาใหม่จากโปรแกรม Microsoft Visual Studio 2010 (Halvorson, 2010) โดยเลือกที่เมนู Start > All Program > Microsoft Visual Studio 2010 > Microsoft Visual Studio 2010 เมื่อเข้าสู่โปรแกรมไมโครซอฟท์วิชวลสตูดิโอ 2010 จะปรากฏหน้าต่างที่มีตัวเลือกดังภาพที่ 4.1 ดังนี้



ภาพที่ 4.1 การเข้าสู่โปรแกรมไมโครซอฟท์วิชวลสตูดิโอ 2010

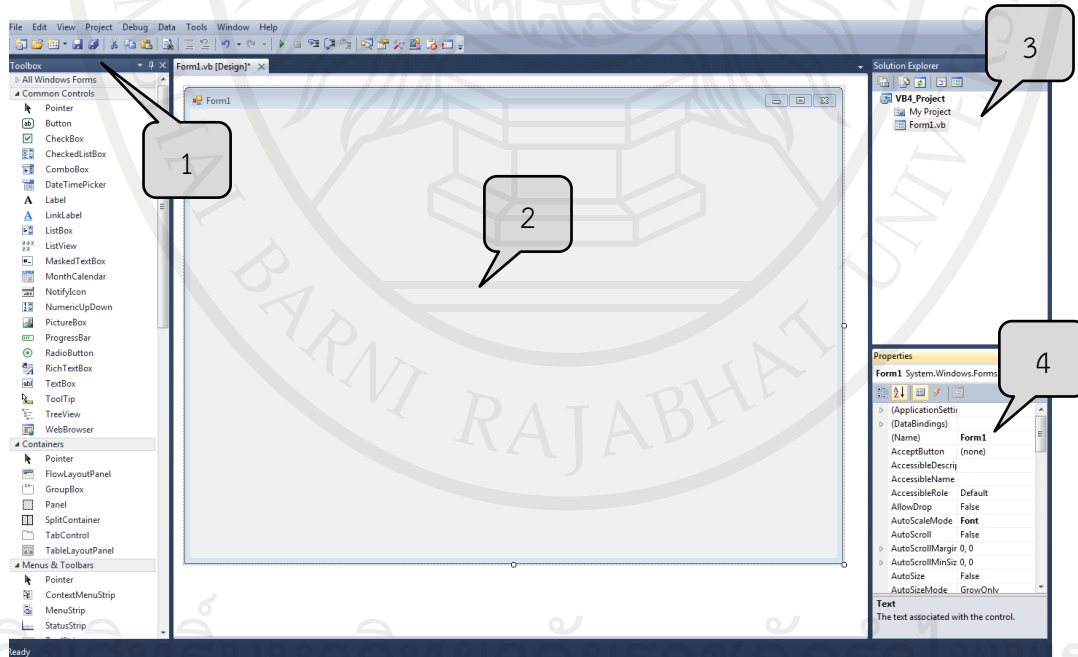
1. New Project... สำหรับ สร้างโครงการใหม่
2. Open Project... เปิดโครงการเก่าที่สร้างไว้แล้ว

ในการสร้างโครงการใหม่ ให้เลือกที่ New Project... จะปรากฏหน้าต่าง New Project ดังภาพที่ 4.2 จากนั้นให้เลือก Visual Basic เลือก Windows Forms Application ตั้งชื่อโครงการในช่อง Name และเลือกปุ่ม Ok จะได้นหน้าต่างที่พร้อมใช้งาน



ภาพที่ 4.2 การสร้างโครงการใหม่

ในหน้าต่างของโครงการจะพบส่วนประกอบต่าง ๆ ของโปรแกรมดังภาพที่ 4.3 ดังนี้ (ธีรวัฒน์, 2554)



ภาพที่ 4.3 หน้าต่างของโครงการ

1. ทุลบ็อกซ์ (Toolbox) เป็นส่วนที่ใช้ในการแสดงปุ่มคอนโทรลและคอมโพเนนต์ต่าง ๆ สำหรับนำมาใช้เป็นส่วนประกอบของฟอร์ม ในการออกแบบเพื่อติดต่อกับผู้ใช้ (User Interface) ซึ่งได้มีการแบ่งออกเป็นกลุ่มย่อย ๆ อีก ซึ่งคอนโทรลแต่ละกลุ่มได้ถูกซ่อนไว้ปรากฏให้เห็นเพียงแค่อีกกลุ่มเท่านั้น ถ้าต้องการแสดงคอนโทรลในกลุ่มใด ให้กดเลือกที่แท็บชื่อกลุ่มนั้นและหากต้องการซ่อนก็ให้กดเลือกที่แท็บชื่อกลุ่มนั้นอีกครั้งหนึ่งและหน้าต่างทุลบ็อกซ์นี้สามารถเรียกใช้ได้ที่เมนู View > Toolbox

2. ฟอร์ม (Form) เป็นส่วนที่ใช้วางปุ่มคอนโทรลต่าง ๆ ที่นี้กออกแบบพัฒนาต้องการ สามารถเรียกใช้ได้ที่เมนู View > Designer โดยการปรับแต่งเช่นการแสดงเส้นกริด เพื่อให้สามารถจัดวางคอนโทรลได้อย่างเป็นระเบียบร้อยนั้นสามารถทำได้โดยเลือกเมนู Tool > Options...

- เลือก Windows Forms Designer > General
- ที่ Layoutmode เลือก Snaptogrid
- Showgrid เป็น True
- Snaptogrid เป็น True

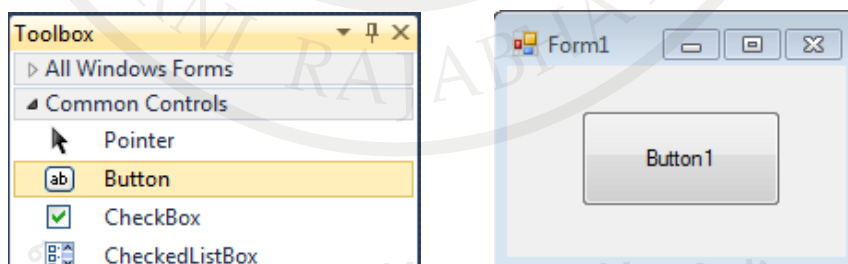
3. โซลูชัน เอ็กซ์พลอยเลอร์ (Solution explorer) เป็นส่วนที่แสดงองค์ประกอบต่าง ๆ ที่มีอยู่ในโปรเจกต์ ซึ่งสามารถดูรายละเอียดของโปรเจกต์ ใช้สลับระหว่างมุมมองการออกแบบและโค้ดของโปรแกรมได้ สามารถเรียกใช้ได้ที่ เมนู View > Solution Explorer

4. หน้าต่างคุณสมบัติ (Properties windows) เป็นส่วนที่ใช้ในการแสดงคุณสมบัติของคอนโทรลหรือคอมโพเนนต์ต่าง ๆ ที่เราเลือก ซึ่งคุณสมบัติเหล่านี้จะแตกต่างกันไปตามคอนโทรลแต่ละชนิด เช่น ขนาด, ตำแหน่ง, สีพื้นหลัง, ลักษณะตัวอักษร, ข้อความบนคอนโทรลนั้น ฯลฯ สามารถเรียกใช้ได้ที่เมนู View > Properties Window

โดยเครื่องมือช่วยออกแบบหน้าจอเบื้องต้นจะอยู่ในกลุ่มทุลบ็อกซ์ แต่ละอันมีหน้าที่และองค์ประกอบดังนี้ (สัจจะ, 2554).

คอนโทรล баттон

ปุ่มคอนโทรล баттон (Control button) ดังภาพที่ 4.4 เป็นปุ่มกดสำหรับการทำงานบางอย่างที่ตอบสนองการกดเมาส์ของผู้ใช้ จัดได้ว่าเป็นคอนโทรลที่มีการใช้งานมากที่สุด เมื่อนำปุ่มคำสั่งมาวางบนฟอร์ม โปรแกรมจะตั้งชื่อเพื่ออ้างถึงปุ่มว่า баттонหนึ่ง (Button1) และเมื่อนำปุ่มมาวางเพิ่ม โปรแกรมจะตั้งชื่อเพื่ออ้างถึงปุ่มว่า баттонสอง (Button2) เช่นนี้ต่อไปเรื่อย ๆ ตามลำดับ

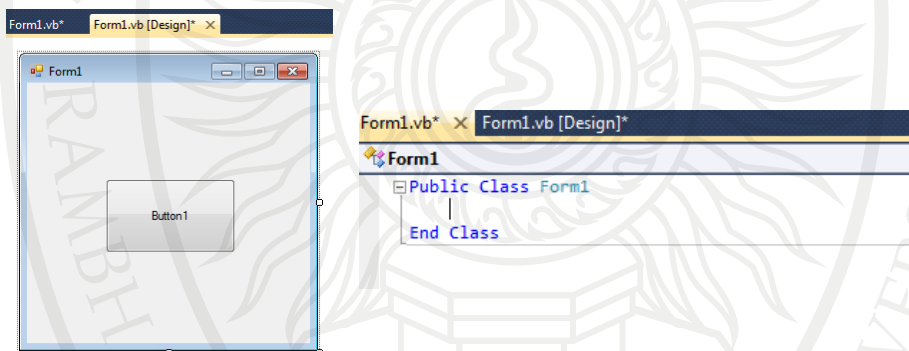


ภาพที่ 4.4 คอนโทรล баттон

ซึ่งปุ่มคอนโทรลบนนี้มีคุณสมบัติที่น่าสนใจดังนี้

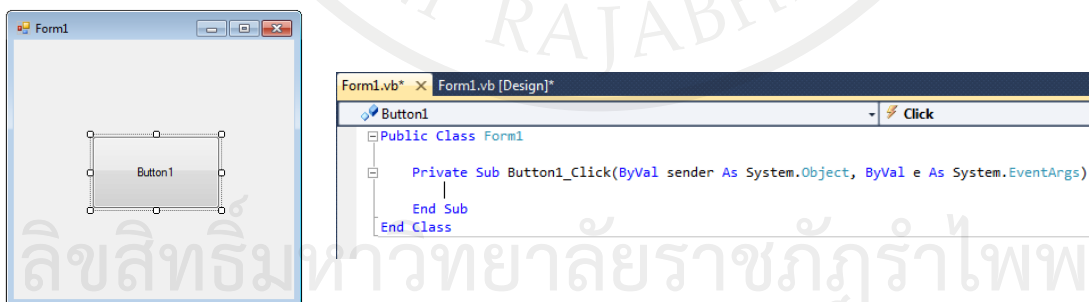
ชื่อคุณสมบัติ	คำอธิบาย
Image	กำหนดรูปภาพที่แสดงบนปุ่มคำสั่ง
BackgroundImage	กำหนดรูปภาพที่แสดงเป็นพื้นหลังของปุ่มคำสั่ง
ImageAlign	กำหนดตำแหน่งที่จะแสดงรูปภาพ
Text	กำหนดข้อความที่แสดงบนปุ่มคำสั่ง
TextAlign	กำหนดตำแหน่งที่จะแสดงข้อความ
Font	กำหนดรูปแบบฟอนต์ที่แสดงบนปุ่มคำสั่ง
Enabled	กำหนดให้ปุ่มคำสั่งปุ่มนี้ใช้งานได้หรือไม่
Visible	กำหนดให้ปุ่มคำสั่งปุ่มนี้สามารถมองเห็นได้ในตอนทำงานหรือไม่

การเขียนโค้ดโปรแกรมหลังจากที่ออกแบบหน้าต่างโปรแกรมเรียบร้อยแล้ว จะทำการเพิ่มชุดคำสั่งให้ออบเจ็กต์แต่ละตัวทำงานตามชุดคำสั่ง ซึ่งการเขียนโค้ดนั้นสามารถทำได้โดยคลิกที่ Form1.vb ตามภาพที่ 4.5 เพื่อเข้าหน้าต่างโค้ดเอดิเตอร์ (Code Editor) ซึ่งสามารถเขียนโค้ดให้กับออบเจ็กต์ได้ตรงเคอร์เซอร์ (Cursor)



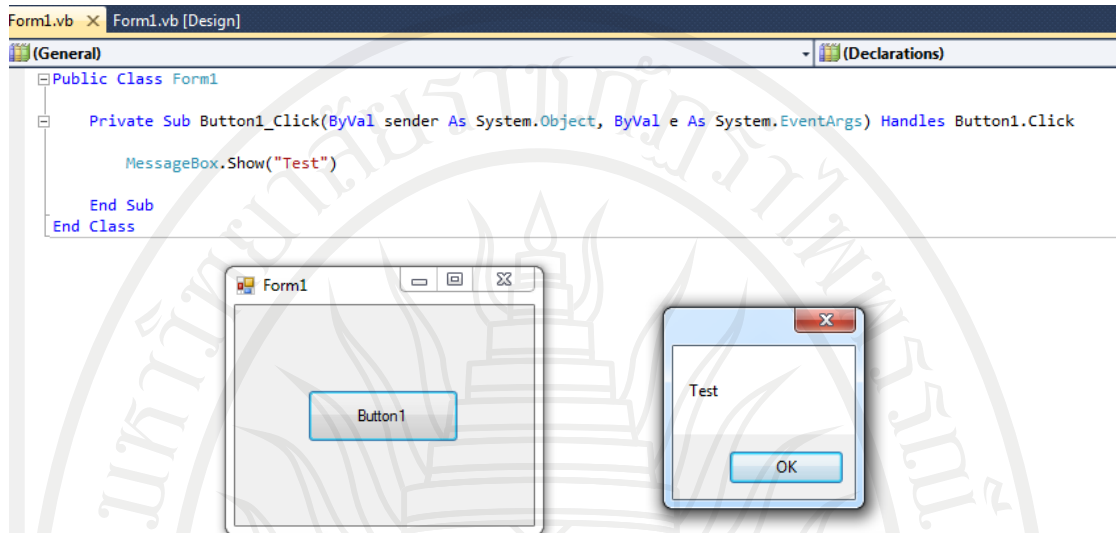
ภาพที่ 4.5 หน้าต่างสำหรับการเขียนคำสั่ง

หรือดับเบิลคลิกตรง “Button 1” เพื่อเขียนโค้ดลงไป ดังภาพที่ 4.6



ภาพที่ 4.6 การเขียนโค้ดลงที่ปุ่มบัตตอน

ทดสอบโดยการเขียน `MessageBox.Show ("Test")` แล้วกด F5 กดปุ่ม Button 1 จะแสดงกล่องข้อความดังภาพที่ 4.7



ภาพที่ 4.7 ตัวอย่างการทดสอบเขียนโปรแกรม

จากภาพที่ 4.7 เป็นการแสดงข้อความการแจ้งข้อมูลให้แก่ผู้ใช้ เช่น ผลลัพธ์ หรือ คำเตือน โดย `MessageBox` มีรูปแบบการใช้งานอย่างง่าย สำหรับศึกษาข้อมูลชนิดโครงสร้างไฟล์ ตัวแปรและการกำหนดค่าต่าง ๆ ต่อไป ดังนี้

รูปแบบ

`MsgBox ("ข้อความ")`

หรือ

`MessageBox.Show ("ข้อความที่ต้องการแสดง", "ข้อความบน TitleBar", MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton)`

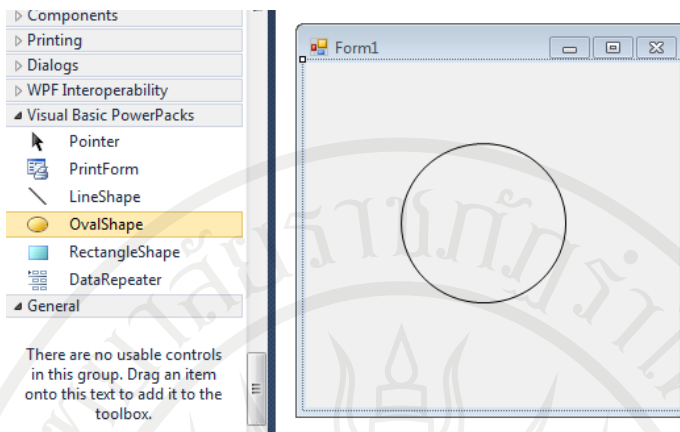
เช่น

`MessageBox.Show ("ต้องการออกจากโปรแกรมใช่หรือไม่?", "ออกจากโปรแกรม", MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton._Button1)`

คอนโทรลโอวัลเชพ (Control ovalshape)

คอนโทรลโอวัลเชพดังภาพที่ 4.8 เป็นเครื่องมือสำเร็จที่คอยช่วยสร้างวงกลมวงรี ซึ่งสามารถนำมาใช้ช่วยแสดงผลเป็นแอลอีดี (LED) ได้

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



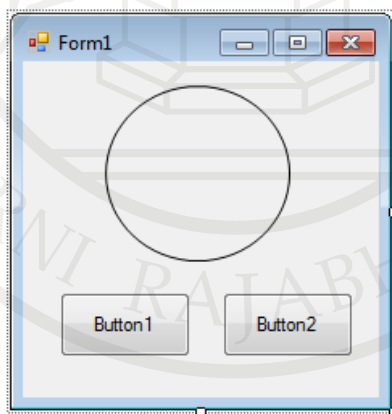
ภาพที่ 4.8 คอนโทรลโอวัลเชพ

ซึ่งมีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
FillStyle	กำหนดรูปภาพที่แสดงบนปุ่มคำสั่ง
FillColor	กำหนดรูปภาพที่แสดงเป็นพื้นหลังของปุ่มคำสั่ง
Location	บอกตำแหน่งแกน X, Y
Size	บอกขนาดความกว้างและความสูง
Visible	กำหนดให้สามารถมองเห็นได้ในตอนทำงานหรือไม่

ตัวอย่าง 4.1 การสร้างปุ่มกดควบคุมหลอดไฟ ติด - ดับ

จุดประสงค์ เพื่อศึกษาการสร้างปุ่มกด ด้วยคอนโทรลปุ่มทอน, คอนโทรลโอวัลเชพและกำหนดค่าคุณสมบัติ



ภาพที่ 4.9 การสร้างปุ่มกด ควบคุมหลอดไฟ ติด - ดับ

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

คำสั่ง มอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ และตามภาพที่ 4.9
2. เขียนโปรแกรมให้กับ OvalShape1 แล้วกำหนดคุณสมบัติดังนี้
 - 2.1 เมื่อเริ่มต้นให้ OvalShape 1 มี FillStyle = FillStyle.Solid
 - 2.2 เมื่อเริ่มต้นให้ OvalShape 1 เป็นสีดำ
3. เขียนโปรแกรมให้กับคอนโทรลботthonให้ทำ งานดังนี้
 - 3.1 เมื่อคลิกวัตถุ Button 1 แล้วทำให้วัตถุ OvalShape 1 เปลี่ยนสีจากสีดำเป็นสีแดง
 - 3.2 เมื่อคลิกวัตถุ Button 2 แล้วทำให้วัตถุ OvalShape 1 เปลี่ยนสีจากรีแดงเป็นสีดำ

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
OvalShape1.FillStyle = PowerPacks.FillStyle.Solid
```

```
OvalShape1.FillColor = Color.Black
```

```
Button1.Text = "เปิด"
```

```
Button2.Text = "ปิด"
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
OvalShape1.FillColor = Color.Red
```

```
End Sub
```

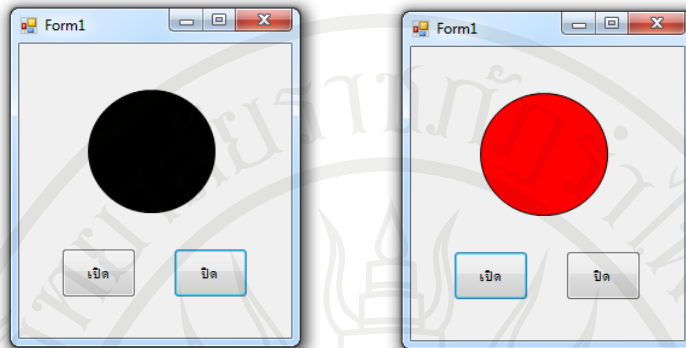
```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
OvalShape1.FillColor = Color.Black
```

```
End Sub
```

```
End Class
```

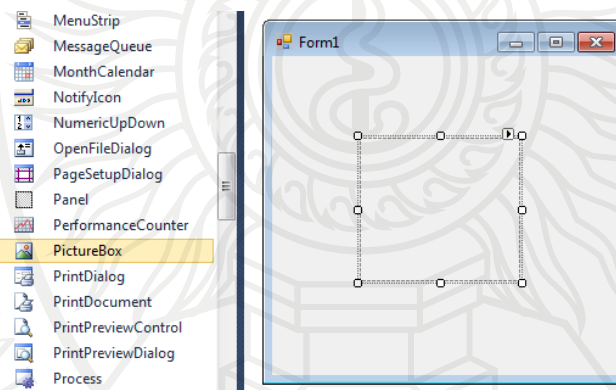
เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.10



ภาพที่ 4.10 ผลลัพธ์การสร้างปุ่มกด ควบคุมหลอดไฟ ติด - ดับ

คอนโทรลพิกเจอร์บ็อกซ์

คอนโทรลพิกเจอร์บ็อกซ์ (Control pictureBox) ดังภาพที่ 4.11 ใช้เพื่อแสดงรูปภาพที่เป็นภาพนิ่ง ซึ่งสามารถจัดการกับภาพที่แสดงได้ในหลายลักษณะ



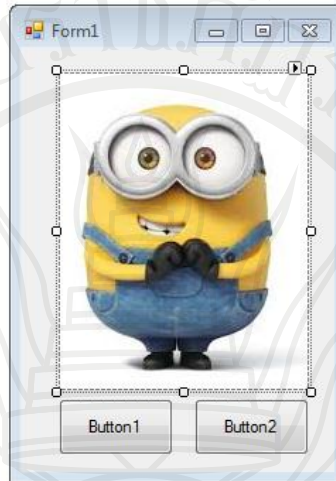
ภาพที่ 4.11 คอนโทรลพิกเจอร์บ็อกซ์

ซึ่งคอนโทรลพิกเจอร์บ็อกซ์มีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
Width	ความกว้างของคอนโทรลพิกเจอร์บ็อกซ์
Hight	ความสูงของคอนโทรลพิกเจอร์บ็อกซ์
SizeMode	การจัดการกับขนาดของภาพ ในกรณีที่ขนาดภาพไม่พอดีกับขนาดของคอนโทรลพิกเจอร์บ็อกซ์
Image	เป็นออบเจ็กต์ที่ใช้จัดการการแสดงผลภาพ ควบคุมกับคอนโทรลพิกเจอร์บ็อกซ์
ImageLocation	เป็นชื่อของไฟล์ ที่เก็บไฟล์ที่จะมาแสดงผล

ตัวอย่าง 4.2 การสร้างปุ่มกดควบคุมมองเห็น - มองไม่เห็น

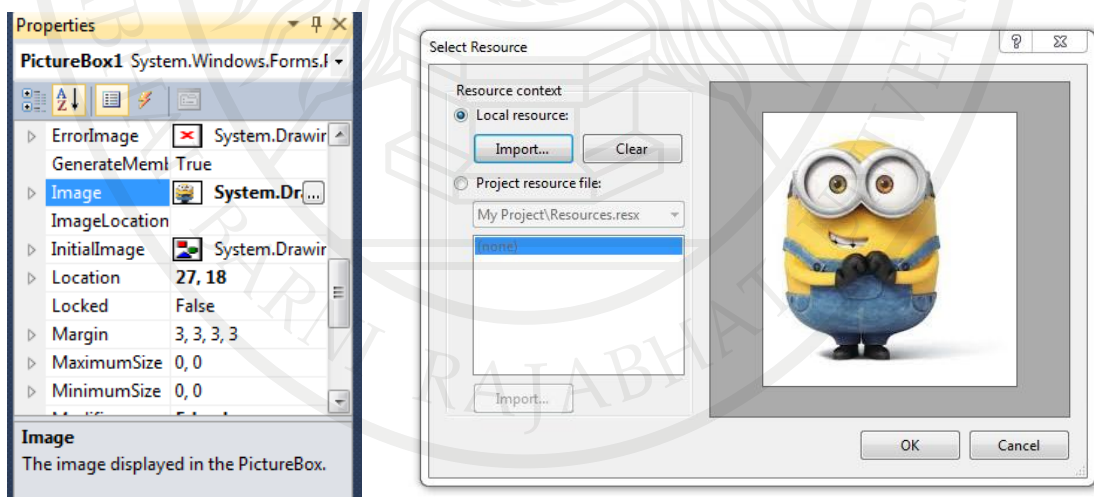
จุดประสงค์ เพื่อศึกษาการสร้างปุ่มกด ด้วยคอนโทรลบัตตอน, คอนโทรลฟิกเจอร์บ็อกซ์ และ กำหนดค่าคุณสมบัติ



ภาพที่ 4.12 การสร้างปุ่มกด ควบคุมมองเห็น - มองไม่เห็น

คำสั่งมอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ Button1 Button2 และ PictureBox1 ตั้งภาพที่ 4.12 และกำหนดคุณสมบัติโดยเลือกรูปภาพจากนั้นหาภาพที่ต้องการตั้งภาพที่ 4.13



ภาพที่ 4.13 การเลือกรูปภาพของคอนโทรลฟิกเจอร์บ็อกซ์

2. เขียนโปรแกรมให้กับวัตถุ Button1 และ Button2 ให้ทำงานดังนี้
 - 2.1 เมื่อคลิกวัตถุ Button 1 แล้วทำให้วัตถุ PictureBox1 มองไม่เห็น
 - 2.2 เมื่อคลิกวัตถุ Button 2 แล้วทำให้วัตถุ PictureBox1 มองเห็น

โปรแกรมคำสั่ง

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Button1.Text = "ปิด"

Button2.Text = "เปิด"

PictureBox1.Visible = False

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

PictureBox1.Visible = False

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

PictureBox1.Visible = True

End Sub

End Class

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.14



ภาพที่ 4.14 ผลลัพธ์การสร้างปุ่มกดควบคุมมองเห็น - มองไม่เห็น

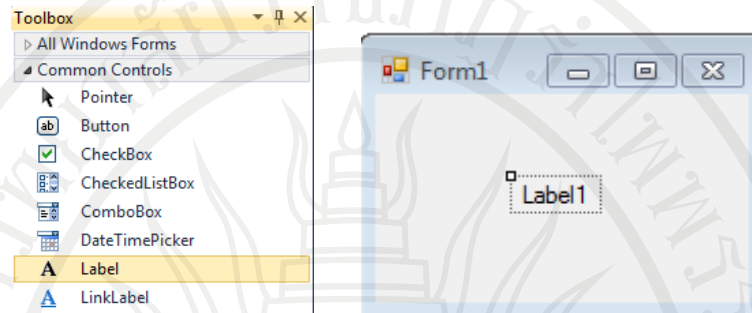
คอนโทรลลาเบล

คอนโทรลลาเบล (Control label) ดังภาพที่ 4.15 ใช้สำหรับแสดงข้อความบนฟอร์มเช่น แสดงผลลัพธ์ ฯลฯ สามารถใช้งานได้โดยการนำคอนโทรลลาเบลมาวางบนฟอร์มแล้วกำหนดข้อความที่ต้องการให้แสดงโดยใช้ Properties Text หรือ ผ่านทาง Properties Windows

เช่น ใส่ข้อความ Label1.Text = “Hello”

หรือกำหนดสี Label1.ForeColor = Color.Red

เมื่อนำลาเบลมาวางบนฟอร์ม โปรแกรมจะตั้งชื่อเพื่ออ้างถึงลาเบลแรกว่า Label1 และเมื่อนำลาเบลมาวางเพิ่ม โปรแกรมจะตั้งชื่อเพื่ออ้างถึงปุ่มว่า Label1 เช่นนี้ต่อไปเรื่อย ๆ ตามลำดับ



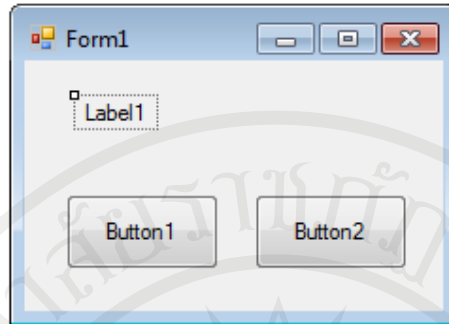
ภาพที่ 4.15 คอนโทรลลาเบล

ซึ่งคอนโทรลลาเบลมีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
AutoSize	ปกติแล้วลาเบลจะมีขนาดพอดีกับข้อความและไม่สามารถใช้เมาส์ลากปรับขนาดได้ เพราะ Properties AutoSize มีค่าเป็น True หากต้องการปรับขนาดตามความต้องการได้ ต้องเปลี่ยน Properties AutoSize เป็น False
BorderStyle	กำหนดว่าจะให้มีเส้นขอบหรือไม่ รวมทั้งกำหนดรูปแบบของเส้นขอบไว้ด้วย
ForeColor	กำหนดสี
Image	กำหนดรูปภาพที่แสดงบนลาเบล
ImageAlign	กำหนดตำแหน่งที่จะแสดงรูปภาพบนลาเบล
Text	กำหนดข้อความที่แสดงบนลาเบล
TextAlign	กำหนดตำแหน่งที่จะแสดงข้อความ
Visible	กำหนดให้สามารถมองเห็นได้ในตอนทำงานหรือไม่

ตัวอย่าง 4.3 การสร้างปุ่มกดควบคุมการแสดงผลข้อความ

จุดประสงค์เพื่อศึกษาการสร้างปุ่มกด ด้วยคอนโทรลปุ่ม,คอนโทรลลาเบลและกำหนดค่าคุณสมบัติ



ภาพที่ 4.16 การสร้างปุ่มกด ควบคุมการแสดงผลข้อความ

คำสั่งมอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ Button1, Button2 และ Label1 ดังภาพที่ 4.16 และกำหนดคุณสมบัติ

2. เขียนโปรแกรมให้กับวัตถุ Button1 และ Button2 ให้ทำงานดังนี้

2.1 เมื่อคลิกวัตถุ Button1 แล้วทำให้ Label1 แสดงข้อความ "Hello" สีขาวและพื้นหลังสีดำ

2.2 เมื่อคลิกวัตถุ Button2 แล้วทำให้ Label1 แสดงข้อความ "Hello" สีดำและพื้นหลังสีขาว

โปรแกรมคำสั่ง

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label1.Visible = False

Button1.Text = "Step 1"

Button2.Text = "Step 2"

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Label1.Visible = True

Label1.Text = "Hello"

Label1.ForeColor = Color.White

Label1.BackColor = Color.Black

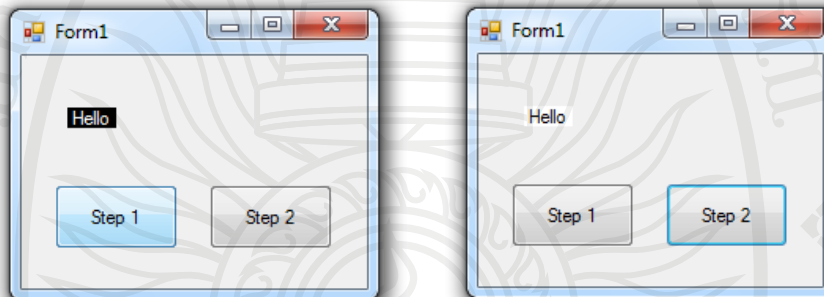
End Sub


```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button2.Click
    Label1.Visible = True
    Label1.Text = "Hello"
    Label1.ForeColor = Color.Black
    Label1.BackColor = Color.White
End Sub
End Class

```

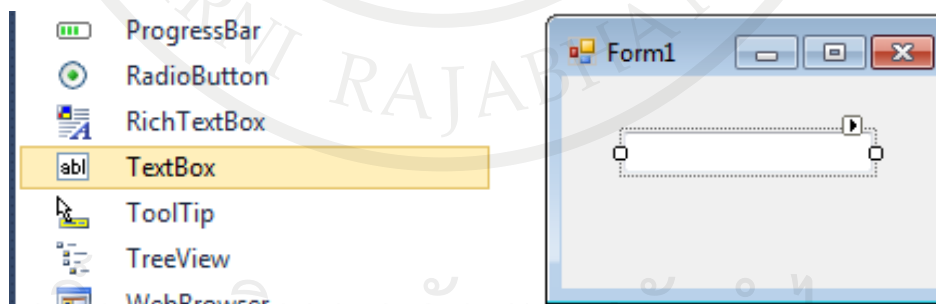
เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.17



ภาพที่ 4.17 ผลลัพธ์การสร้างปุ่มกดควบคุมการแสดงข้อความ

คอนโทรลเท็กซ์บ็อกซ์

คอนโทรลเท็กซ์บ็อกซ์ (Control textbox) ดังภาพที่ 4.18 เป็นคอนโทรลที่ใช้งานบ่อยมากในการรับข้อมูลจากผู้ใช้เพื่อนำไปประมวลผลต่อไป นอกจากนี้ยังใช้ในการแสดงข้อความได้อีกด้วย คอนโทรลเท็กซ์บ็อกซ์สามารถใช้ได้ทั้งแบบรับข้อมูลบรรทัดเดียวและแบบหลายบรรทัดได้ โดยถ้าต้องการให้รับข้อมูลได้หลายบรรทัดจะต้องกำหนดที่ Properties MultiLines ของคอนโทรลเท็กซ์บ็อกซ์ให้เป็น True (ค่าเริ่มต้นเป็น False)



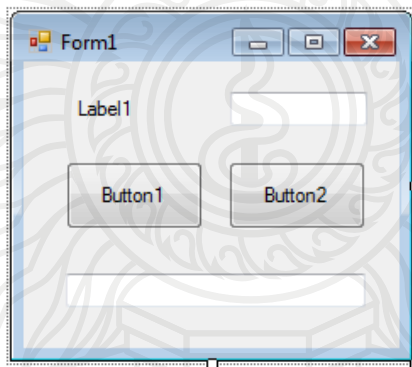
ภาพที่ 4.18 คอนโทรลเท็กซ์บ็อกซ์

ซึ่งคอนโทรลเท็กซ์บ็อกซ์มีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
MaxLength	กำหนดความยาวสูงสุดของข้อความที่สามารถพิมพ์ได้
MultiLine	กำหนดให้แสดงและรับข้อความได้หลายบรรทัด
ScrollBars	กำหนดให้แสดงสกรอลบาร์
TextAlign	กำหนดตำแหน่งที่จะแสดงข้อความ
ReadOnly	กำหนดให้อ่านได้อย่างเดียวแก้ไขไม่ได้
PasswordChar	สร้างเท็กซ์บ็อกซ์แบบรหัสผ่าน
TextChanged	กำหนดให้มีการเปลี่ยนแปลงเกิดขึ้นกับข้อความในเท็กซ์บ็อกซ์ เช่น การพิมพ์ หรือ การลบข้อความ

ตัวอย่าง 4.4 การสร้างวัตถุเท็กซ์บ็อกซ์การป้อนข้อมูลและข้อความ

จุดประสงค์เพื่อศึกษาการสร้างและการใช้งานของวัตถุป้อนข้อมูลและข้อความด้วยคอนโทรลปุ่มทอน, ลาเบล, เท็กซ์บ็อกซ์และกำหนดค่าคุณสมบัติ



ภาพที่ 4.19 การสร้างคอนโทรลเท็กซ์บ็อกซ์สำหรับการป้อนข้อมูลและข้อความ

คำสั่งมอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ Button1, Button2, Label1, TextBox1 และ TextBox2 ดังภาพที่ 4.19
2. เมื่อเริ่มรันโปรแกรมให้
 - 2.1 Label1 แสดงข้อความว่า "กรุณาบอกชื่อเล่น" ด้วยตัวอักษรสีดำ
 - 2.2 Button1 แสดงข้อความว่า "หักทนาย"
 - 2.3 Button2 แสดงข้อความว่า "ลบ"
3. พิมพ์ชื่อเราไว้ที่ TextBox1
4. เขียนโปรแกรมให้กับวัตถุ Button1 และ Button2 ให้ทำงานดังนี้

4.1 เมื่อคลิกวัตถุ Button1 แล้วทำให้ TextBox2 แสดงผลว่า "สวัสดีครับ คุณ " ตามด้วยชื่อเราที่พิมพ์ใน TextBox1 และต่อด้วยคำว่า " สบายดีนะครับ" ดังภาพที่ 4.20

4.2 เมื่อคลิกวัตถุ Button2 แล้วทำให้ข้อความใน TextBox1 และ TextBox2 หายไป

โปรแกรมคำสั่ง

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label1.Text = "กรุณามอบชื่อเล่น"

Label1.ForeColor = Color.Black

Button1.Text = "พักทหาย"

Button2.Text = "ลบ"

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

TextBox2.Text = "สวัสดีครับ คุณ " & TextBox1.Text & " สบายดีนะครับ"

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

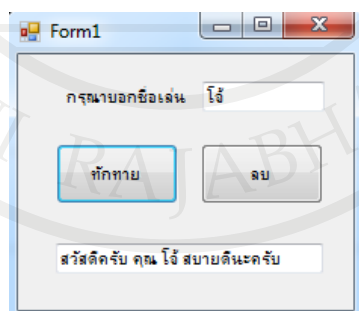
TextBox1.Text = " "

TextBox2.Text = " "

End Sub

End Class

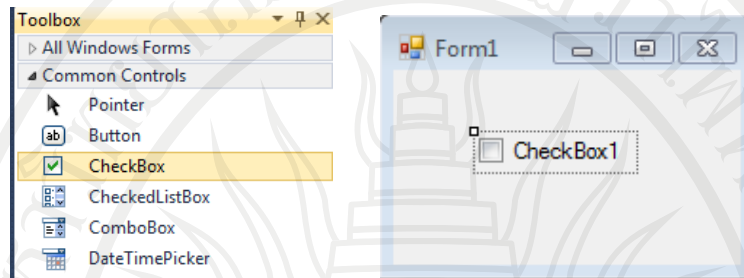
เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.20



ภาพที่ 4.20 ผลลัพธ์การแสดงผลการสร้างคอนโทรลที่กซ์บ็อกซ์สำหรับการป้อนข้อมูลและข้อความ

คอนโทรลเช็คบ็อกซ์

คอนโทรลเช็คบ็อกซ์ (Control checkbox) ดังภาพที่ 4.21 เป็นคอนโทรลสำหรับสร้างรายการตัวเลือก เพื่อให้ผู้ใช้สามารถเลือกสิ่งที่ต้องการ แทนการคีย์ข้อมูล โดยผู้ใช้ ซึ่งสามารถใช้เลือกมากกว่า 1 รายการ การใช้คอนโทรลเช็คบ็อกซ์นั้น เนื่องจากว่าคอนโทรลเช็คบ็อกซ์แต่ละตัวที่วางบนฟอร์มนั้นจะเป็นอิสระจากกัน ดังนั้นการตรวจสอบว่าตัวใดถูกเลือกจะต้องตรวจสอบทีละตัวจนครบ



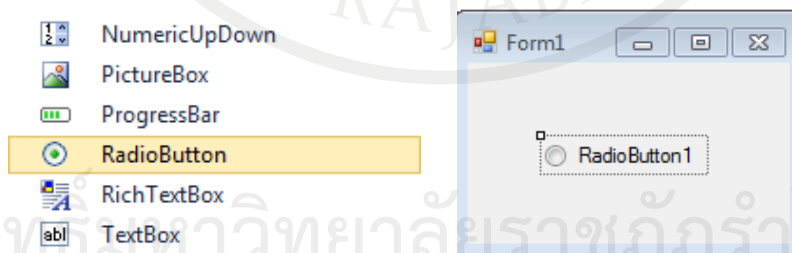
ภาพที่ 4.21 คอนโทรลเช็คบ็อกซ์

ซึ่งคอนโทรลเช็คบ็อกซ์มีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
Checked	เป็นการตรวจสอบว่าได้เลือกที่ตัวเลือกนี้หรือไม่ หากเลือกจะมีค่าเป็น True หรือกำหนดให้ตัวเลือกนั้นถูกเลือกหรือไม่เลือก
Text	กำหนดข้อความที่อยู่บนคอนโทรลเช็คบ็อกซ์
Unchecked	เป็นการตรวจสอบว่าได้เลือกที่ตัวเลือกนี้หรือไม่ หากเลือกจะมีค่าเป็น False หรือกำหนดให้ตัวเลือกนั้นถูกเลือกหรือไม่เลือก

คอนโทรลเรดิโอบัตตอน

คอนโทรลเรดิโอบัตตอน (Control radiobutton) ดังภาพที่ 4.22 เป็นคอนโทรลสำหรับเลือกรายการเหมือนกับคอนโทรลเช็คบ็อกซ์แต่ต่างกันตรงที่คอนโทรลเรดิโอบัตตอนจะสามารถเลือกได้เพียงรายการเดียวเท่านั้น โดยในการใช้งานคอนโทรลเรดิโอบัตตอนที่มีข้อมูลหลาย ๆ กลุ่ม และจำเป็นต้องเลือกข้อมูลจากทุกกลุ่ม จะไม่สามารถทำได้เพราะโปรแกรมจะถือว่าเป็นกลุ่มเดียวกัน ดังนั้น แนวทางแก้ไขสามารถทำได้โดยใช้กรุปบ็อกซ์ในการจัดกลุ่มเสียก่อน



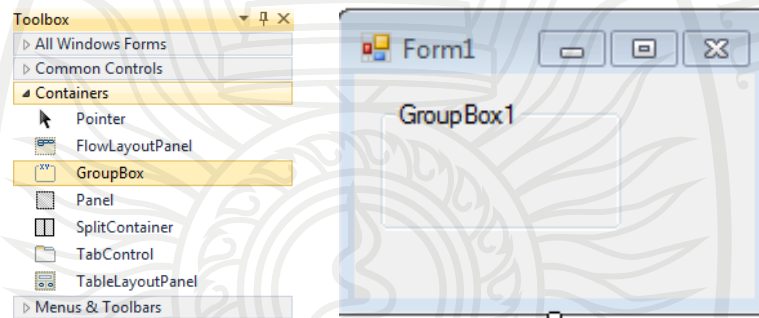
ภาพที่ 4.22 คอนโทรลเรดิโอบัตตอน

โดยคอนโทรลเรดิโอบัตทอนมีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
Checked	เป็นการตรวจสอบว่าได้เลือก ที่ตัวเลือกนี้หรือไม่ หากเลือกจะมีค่าเป็น True หรือกำหนดให้ตัวเลือกนั้นถูกเลือกหรือไม่เลือก
Text	กำหนดข้อความที่อยู่บนคอนโทรลเรดิโอบัตทอน
Unchecked	เป็นการตรวจสอบว่าได้เลือกที่ตัวเลือกนี้หรือไม่ หากเลือกจะมีค่าเป็น False หรือกำหนดให้ตัวเลือกนั้นถูกเลือกหรือไม่เลือก

คอนโทรลกรุปบ็อกซ์

คอนโทรลกรุปบ็อกซ์ (Control groupbox) ดังภาพที่ 4.23 เป็นคอนโทรลเพื่อจัดกลุ่ม ซึ่งต้องวางกรุปบ็อกซ์ลงไปก่อน แล้วค่อยวางคอนโทรลที่จะจัดกลุ่มลงไป



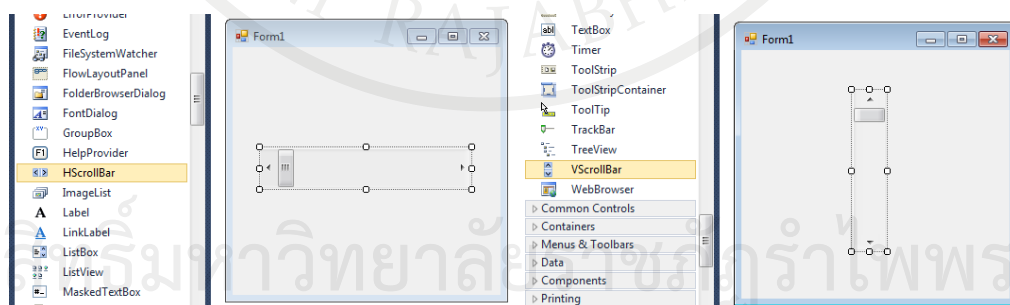
ภาพที่ 4.23 คอนโทรลกรุปบ็อกซ์

โดยคอนโทรลกรุปบ็อกซ์มีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
Text	ข้อความที่แสดงชื่อกลุ่ม ซึ่งจะปรากฏที่มุมบนซ้ายของกรุปบ็อกซ์

คอนโทรลสโครลบาร์

คอนโทรลสโครลบาร์ (Control scroll bar) ดังภาพที่ 4.24 มีทั้งแบบแนวนอน เอชสโครลบาร์ (HScroll bar) และแนวตั้ง วีสโครลบาร์ (VScroll bar) ขึ้นอยู่กับตำแหน่ง และการนำไปใช้งานของผู้ใช้



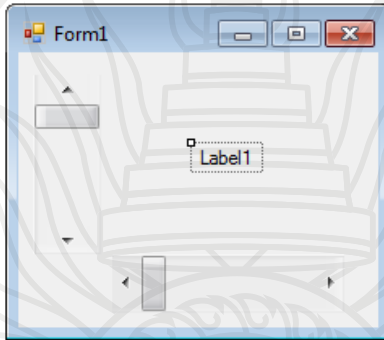
ภาพที่ 4.24 คอนโทรลสโครลบาร์

โดยสโครลบาร์มีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
Maximum	ปริมาณสูงสุดของสโครลบาร์
Minimum	ปริมาณต่ำสุดของสโครลบาร์

ตัวอย่าง 4.5 การสร้างและใช้งานคอนโทรลสโครลบาร์

จุดประสงค์เพื่อศึกษาการสร้างและการใช้งานคอนโทรลเอสโครลบาร์และวีลสโครลบาร์ และกำหนดค่าคุณสมบัติ



ภาพที่ 4.25 การสร้างและใช้งานคอนโทรลสโครลบาร์

คำสั่งมอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ Label1, HScrollBar1 และ VScrollBar1 ดังภาพที่ 4.25
2. เมื่อเริ่มรันโปรแกรมให้
 - 2.1 Label1 แสดงข้อความว่า "0.00 V" ด้วยตัวอักษรสีดำ และพื้นหลังสีเหลือง
 - 2.2 HScrollBar1 มีค่าสูงสุดที่ 200 และต่ำสุดที่ 1
 - 2.3 VScrollBar1 มีค่าสูงสุดที่ 200 และต่ำสุดที่ 1
3. เมื่อมีการเคลื่อน HScrollBar1 หรือ VScrollBar1 ให้ Label1 นำค่า VScrollBar1/HScrollBar1 มาแสดงผล

โปรแกรมคำสั่ง

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label1.BackColor = Color.Yellow

Label1.Text = "0.00 V"

HScrollBar1.Maximum = 200

HScrollBar1.Minimum = 1

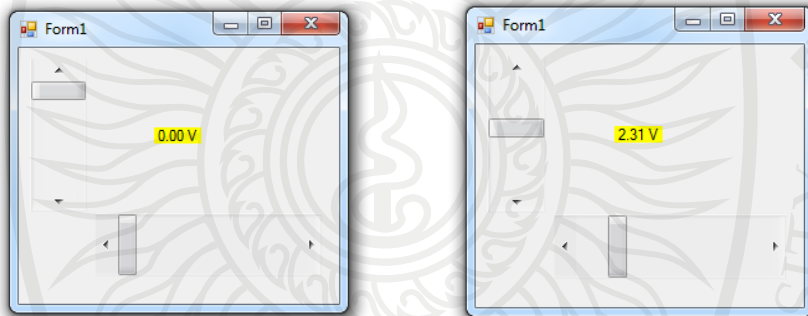
VScrollBar1.Maximum = 200

```

VScrollBar1.Minimum = 1
End Sub
Private Sub HScrollBar1_Scroll(ByVal sender As System.Object, ByVal e As _
System.Windows.Forms.ScrollEventArgs) Handles HScrollBar1.Scroll
    Label1.Text = Format((VScrollBar1.Value / HScrollBar1.Value), "0.00 V")
End Sub
Private Sub VScrollBar1_Scroll(ByVal sender As System.Object, ByVal e As _
System.Windows.Forms.ScrollEventArgs) Handles VScrollBar1.Scroll
    Label1.Text = Format((VScrollBar1.Value / HScrollBar1.Value), "0.00 V")
End Sub
End Class

```

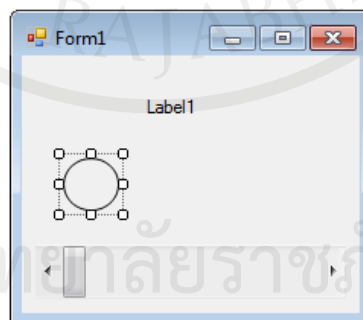
เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.26



ภาพที่ 4.26 ผลลัพธ์การสร้างและใช้งานคอนโทรลสโครลบาร์

ตัวอย่าง 4.6 การสร้างและใช้งานวัตถุเคลื่อนไหวที่

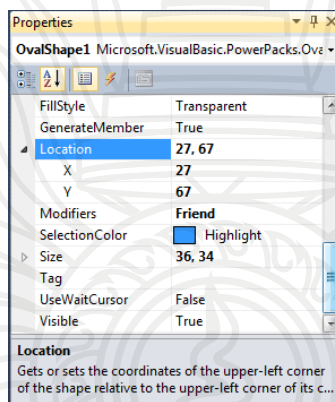
จุดประสงค์เพื่อศึกษาการสร้างและการเขียนโปรแกรมใช้งานให้อิโวลูทเคชั่นที่ และกำหนดค่าคุณสมบัติ



ภาพที่ 4.27 การสร้างและใช้งานวัตถุเคลื่อนไหวที่

คำสั่งมอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ Label1, HScrollBar1 และ OvalShape 1 ดังภาพที่ 4.27
2. เมื่อเริ่มรันโปรแกรมให้
 - 2.1 Label1 แสดงข้อความว่า "0.00 V" ด้วยตัวอักษรสีดำ และพื้นหลังสีเหลือง
 - 2.2 HScrollBar1 มีค่าสูงสุดที่ 200 และต่ำสุดที่ 1
 - 2.3 บันทึกค่า Properties ของ OvalShape1 ในช่องของ Location ดังภาพที่ 4.28 ซึ่งในภาพ X อยู่ที่ 27 และ Y อยู่ที่ 67
3. เมื่อมีการเคลื่อน HScrollBar1 ให้ OvalShape1 เคลื่อนที่ตามตำแหน่ง Location x = HScrollBar1



ภาพที่ 4.28 ค่าคุณสมบัติของโอวีเอลเชพ

โปรแกรมคำสั่ง

Public Class Form1

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Label1.BackColor = Color.Yellow
```

```
Label1.Text = "0.00 V"
```

```
HScrollBar1.Maximum = 200
```

```
HScrollBar1.Minimum = 1
```

```
End Sub
```

```
Private Sub HScrollBar1_Scroll(ByVal sender As System.Object, ByVal e As System.Windows.Forms.ScrollEventArgs) Handles HScrollBar1.Scroll
```

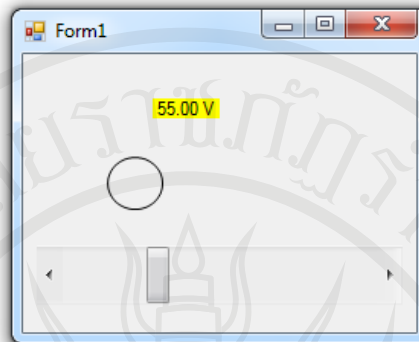
```
Label1.Text = Format((HScrollBar1.Value), "0.00 V")
```

```
OvalShape1.Location = New Point(HScrollBar1.Value, 67)
```

```
End Sub
```

```
End Class
```

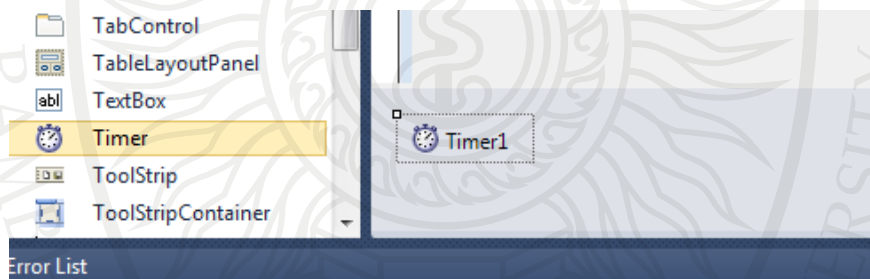

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.29



ภาพที่ 4.29 ผลลัพธ์การสร้างและใช้งานวัตถุ เคลื่อนที่

คอนโทรลไทมเมอร์

คอนโทรลไทมเมอร์ (Control timer) ดังภาพที่ 4.30 มีหน้าที่สำหรับเป็นตัวจับเวลาในการประมวลผลหรือทำหน้าที่หน่วงเวลาหรือตั้งเวลา เพื่อกระทำการอย่างใดอย่างหนึ่ง เมื่อกด F5 เพื่อสั่งงานโปรแกรมจะไม่ปรากฏขึ้นมา แต่เวลาจะเดินไปเรื่อย ๆ จนถึงช่วงเวลาที่ได้โปรแกรมไว้



ภาพที่ 4.30 คอนโทรลไทมเมอร์

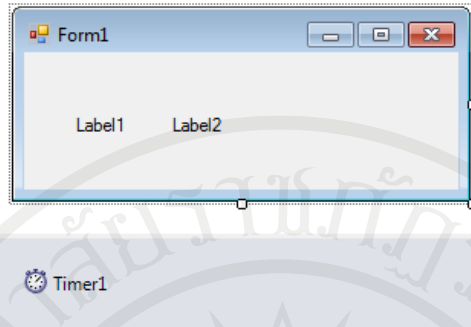
โดยคอนโทรลไทมเมอร์ มีคุณสมบัติที่น่าสนใจดังนี้

ชื่อคุณสมบัติ	คำอธิบาย
Interval	สำหรับกำหนดเวลา เพื่อสั่งให้คอนโทรลไทมเมอร์เริ่มต้นทำงาน
Start	ให้ไทมเมอร์เริ่มทำงาน
Stop	ให้ไทมเมอร์หยุดทำงาน

ตัวอย่าง 4.7 การใช้งานคอนโทรลไทมเมอร์

จุดประสงค์เพื่อศึกษาการสร้างและการเขียนโปรแกรมใช้งานคอนโทรลไทมเมอร์

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 4.31 การเขียนโปรแกรมใช้งานคอนโทรลไทมเมอร์

คำสั่งมอบหมายงาน

1. ให้สร้างฟอร์ม วางวัตถุ Label1, Label2 และ Timer1 ดังภาพที่ 4.31
2. เมื่อเริ่มรันโปรแกรมให้
 - 2.1 Label1 แสดงข้อความว่า "Time"
 - 2.2 Label2 แสดงเวลาปัจจุบัน

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles MyBase.Load
```

```
        Label1.Text = "Time"
```

```
        Timer1.Start()
```

```
    End Sub
```

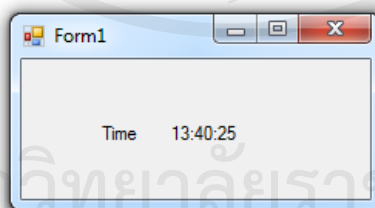
```
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles Timer1.Tick
```

```
        Label2.Text = TimeOfDay
```

```
    End Sub
```

```
End Class
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 4.32



ภาพที่ 4.32 ผลลัพธ์การใช้งานคอนโทรลไทมเมอร์

การตั้งชื่อคอนโทรลและอ็อบเจกต์ตามคำแนะนำของไมโครซอฟท์

เนื่องจากวิซวลเบสิกมีคอนโทรลต่าง ๆ มากมาย โปรแกรมเมอร์หรือผู้เขียนโปรแกรมมักใช้ค่านำหน้า (Prefixes) ในการตั้งชื่อคอนโทรลต่อไปนี้ ซึ่งเป็นคำแนะนำของไมโครซอฟท์ ดังตารางที่ 4.1 ซึ่งไม่ใช่ข้อบังคับที่ผู้เขียนโปรแกรมไม่ต้องใช้ก็ได้

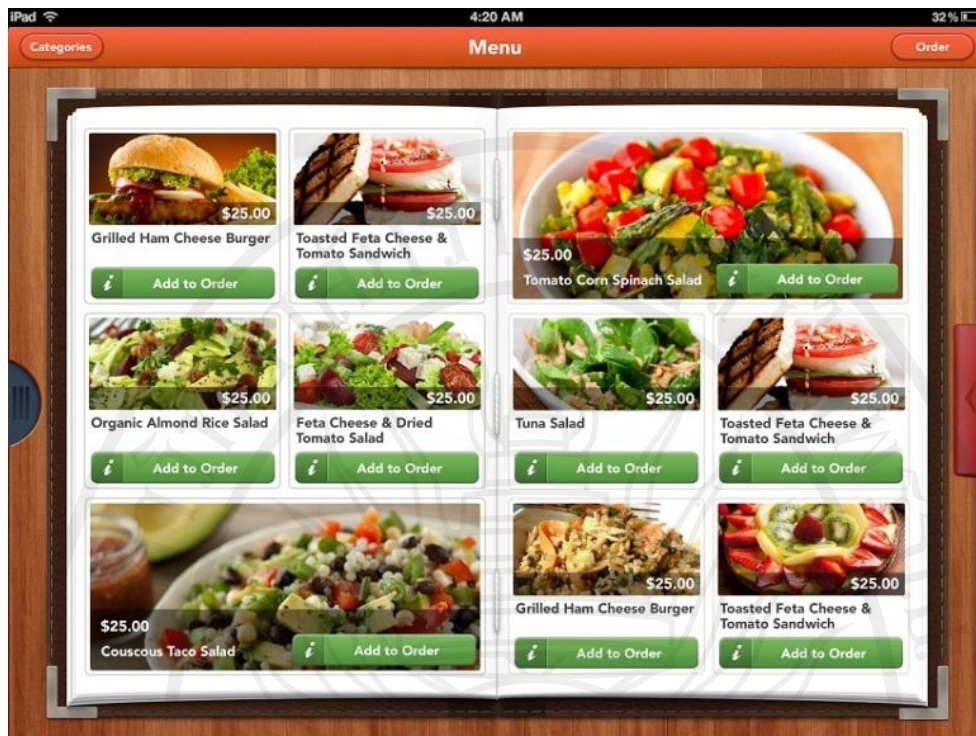
ตารางที่ 4.1 การตั้งชื่อคอนโทรลและอ็อบเจกต์ตามคำแนะนำของไมโครซอฟท์

คอนโทรล	ค่านำหน้า	ตัวอย่าง
Button	btn	btnStart
CheckBox	chk	chkStudents
ComboBox	cbo	cboSalary
CommonDialog	dlg	dlgSave
HScrollBar	hsb	hsbColor
Label	lbl	lblName
ListBox	lst	lstMouth
PictureBox	pic	picPeople
TextBox	txt	txtInput
Timer	tmr	tmrCount
VScrollBar	vsb	vsbHeight

หลังจากการศึกษาแนวทางในการออกแบบหน้าจอและเครื่องมือในการคอนโทรลเบื้องต้น นักศึกษาสามารถนำความรู้มาใช้ในการออกแบบได้หลากหลายตามแต่ลักษณะงานที่พบได้ดังภาพที่ 4.34 และภาพที่ 4.33



ภาพที่ 4.33 ตัวอย่างอุปกรณ์ช่วยนำทาง
ที่มา (Garmin. n.d.)



ภาพที่ 4.34 ตัวอย่างหน้าจอเมนูสินค้าที่สามารถใส่ภาพสินค้าได้เอง
ที่มา (Perfect Solution. n.d.)



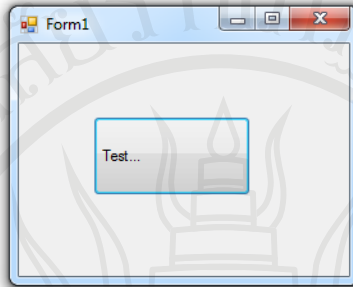
ภาพที่ 4.35 ตัวอย่างหน้าจอสถานะโต๊ะ สำหรับร้านอาหาร
ที่มา (Perfect Solution. n.d.)

สรุป

การทำงานด้านวิศวกรรมนั้น มักเจอปัญหามากหลาย ๆ แบบเพื่อสั่งงานให้อุปกรณ์ต่าง ๆ ทำงานในงานที่ต้องควบคุมผ่านคอมพิวเตอร์นั้น การออกแบบปุ่มกดดี ๆ ที่สามารถแสดงสถานะในการกดได้จึงเป็นสิ่งที่ช่วยในการทำงานได้ดีมาก ตลอดจนการแสดงข้อความเตือนหรืออื่น ๆ การหยิบคอนโทรลเบื้องต้นของวิซวลเบสิกเข้ามาใช้นั้นก็มีด้วยกันหลายแบบหลายชนิด ซึ่งหากแบ่งในส่วนของ การรับข้อมูลเข้า คอนโทรลที่ใช้ได้แก่ คอนโทรลบัตตอน, คอนโทรลเท็กซ์บ็อกซ์, คอนโทรลเช็คบ็อกซ์, คอนโทรลเรดิโอบัตตอน, คอนโทรลกรุปบ็อกซ์หรือคอนโทรลสโครลบาร์ ตลอดจนในส่วนข้อมูลที่ส่งออก คอนโทรลที่ใช้ได้แก่ คอนโทรลโอวิลเซพ, คอนโทรลฟิวเจอร์บ็อกซ์หรือ คอนโทรลลาเบลนอกจากนี้ยังมีคอนโทรลที่คอยช่วยจัดกลุ่มหรือคอยช่วยดำเนินการ ได้แก่ คอนโทรลกรุปบ็อกซ์หรือคอนโทรลไทมเมอร์ ซึ่งแต่ละแบบมีวิธีการเขียนควบคุมแตกต่างกันไม่มาก โดยเฉพาะคุณสมบัติบางตัวของอุปกรณ์อย่าง Text ที่มีไว้กำหนดข้อความบนคอนโทรล สามารถกำหนดได้ที่คุณสมบัติหรือสามารถเขียนเป็นโปรแกรมคำสั่ง ลงไปในฟอร์มเพื่อกำหนดคุณสมบัติได้อีกด้วย

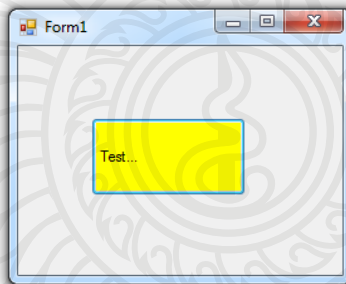
แบบฝึกหัดที่ 4

1. จงสร้างคอนโทรลUTTONขึ้นมา เมื่อกด F5 แล้วให้แสดงผลดังนี้
 - 1.1 ให้คอนโทรลUTTONแสดงข้อความ “Test...” และขีดไปทางด้านซ้าย ดังภาพที่ 4.36



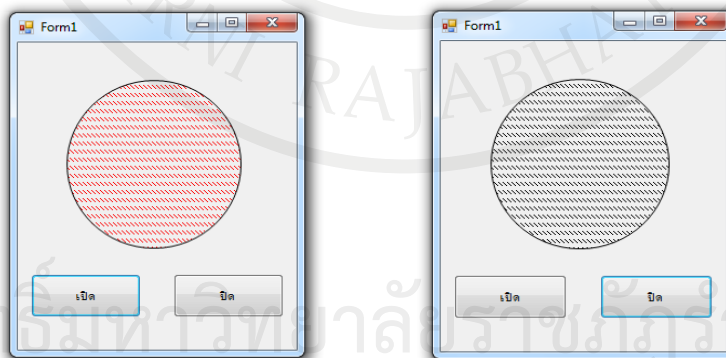
ภาพที่ 4.36 ตัวอย่างคอนโทรลUTTONแสดงข้อความ “Test...”

- 1.2 ให้คอนโทรลUTTONเมื่อถูกกดแล้วกลายเป็นสีเหลือง ดังภาพที่ 4.37



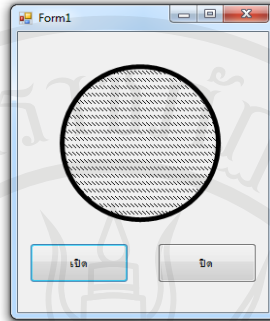
ภาพที่ 4.37 ตัวอย่างคอนโทรลUTTONเมื่อถูกกดแล้วกลายเป็นสีเหลือง

2. จากตัวอย่าง 4.1 การสร้างปุ่มกดควบคุมหลอดไฟ ติด – ดับ หากต้องการให้คอนโทรลไอเวิลเซพแสดงผลภายในเป็นเส้นขีดในแนวนอน ดังภาพที่ 4.38 ด้วยวิธีการเขียนโปรแกรมคำสั่งต้องเขียนอย่างไร



ภาพที่ 4.38 ตัวอย่างคอนโทรลไอเวิลเซพแสดงผลภายในเป็นเส้นขีดในแนวนอน

3. จากตัวอย่าง 4.1 การสร้างปุ่มกดควบคุมหลอดไฟ ติด - ดับ หากต้องการให้เส้นขอบคอนโทรลโอเวิลเซพมีความหนา ดังภาพที่ 4.39 ด้วยวิธีการเขียนโปรแกรมคำสั่งต้องเขียนอย่างไร



ภาพที่ 4.39 ตัวอย่างเส้นขอบคอนโทรลโอเวิลเซพที่มีความหนา

4. จากตัวอย่าง 4.3 การสร้างปุ่มกดควบคุมการแสดงความถี่ หากต้องการให้ขนาดของคอนโทรลลาเบลมีขนาดใหญ่ขึ้น ดังภาพที่ 4.40 ด้วยวิธีการเขียนโปรแกรมคำสั่งต้องเขียนอย่างไร



ภาพที่ 4.40 ตัวอย่างขนาดของคอนโทรลลาเบลมีขนาดใหญ่ขึ้น

5. จงอธิบายโปรแกรมคำสั่งต่อไปนี้ว่าคืออะไร พร้อมทั้งแสดงการนำไปใช้
`Label1.Font = New Drawing.Font("Times New Roman", 16, FontStyle.Bold)`
6. หากต้องการให้คอนโทรลเท็กซ์บ็อกซ์แสดงและรับข้อความได้หลายบรรทัดต้องเขียนโปรแกรมคำสั่งอย่างไร
7. จงออกแบบโปรแกรมที่มีการใช้คอนโทรลสโครลบาร์มา 1 ตัวอย่าง
8. จากตัวอย่าง 4.7 หากเขียนโปรแกรมคำสั่งดังนี้จะเกิดอะไรขึ้น อธิบาย
`Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load
Label1.Text = "Time"
Label2.Text = TimeOfDay
Timer1.Start()
End Sub`

เอกสารอ้างอิง

- ธีรวัฒน์ ประกอบผล, (2554). คู่มือการเขียนแอปพลิเคชันด้วย Visual Basic 2010. กรุงเทพมหานคร: ชิมพลิฟาย.
- สัจจะ จรัสรุ่งรวีวรรณ, (2554). คู่มือเรียนและใช้งาน Visual Basic 2010. นนทบุรี: ไอทีซี พีริเมียร์จำกัด.
- Halvorson, M. (2010). Microsoft Visual Basic 2010 Step by Step. United States of America.
- Garmin (n.d.). On the Road nuvi42 [On-line] Available : <http://www.garmin.co.th/products/ontheroad/nuvi42/>[2556, December 8].
- Perfect Solution (n.d.). ระบบร้านอาหาร โปรแกรมร้านอาหาร POS ร้านอาหาร รองรับคำสั่งด้วย iPOD, iPAD, iPhone ใช้งานง่าย [On-line] Available : <http://pos-restaurantprogram.blogspot.com/2013/06/ipad-take-order.html>[2556, December 8].
- Smashing Magazine (2010). Color Theory For Designers: Creating Your Own Color Palettes. [On-line] Available : <https://www.smashingmagazine.com/2010/02/color-theory-for-designer-part-3-creating-your-own-color-palettes/> [2556, December 8].

แผนบริหารการสอนประจำบทที่ 5

หัวข้อเนื้อหาบทที่ 5

1. ตัวแปร
2. คำสงวน
3. ค่าคงที่
4. ชนิดของข้อมูล
5. การกำหนดค่าให้ตัวแปร
6. โอเปอเรเตอร์
7. ลำดับการทำงานของโอเปอเรเตอร์
8. การเขียนคำอธิบายโปรแกรม
9. คำสั่งในการเขียนโปรแกรม
10. การรับข้อมูลและแสดงข้อความ
11. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 5 แล้วนักศึกษาสามารถ

1. รู้จักตัวแปรในการเขียนโปรแกรมคอมพิวเตอร์ได้
2. รู้จักคำสงวนในการเขียนโปรแกรมคอมพิวเตอร์ได้
3. รู้จักค่าคงที่ในการเขียนโปรแกรมคอมพิวเตอร์ได้
4. รู้จักชนิดของข้อมูลในการเขียนโปรแกรมคอมพิวเตอร์ได้
5. รู้จักและสามารถกำหนดค่าให้ตัวแปรในการเขียนโปรแกรมคอมพิวเตอร์ได้
6. รู้จักโอเปอเรเตอร์ในการเขียนโปรแกรมคอมพิวเตอร์ได้
7. รู้จักลำดับการทำงานของโอเปอเรเตอร์ในการเขียนโปรแกรมคอมพิวเตอร์ได้
8. รู้จักและเขียนคำอธิบายโปรแกรมได้
9. รู้จักและใช้คำสั่งในการเขียนโปรแกรมคอมพิวเตอร์ได้
10. รู้จักการรับข้อมูลและแสดงข้อความในการเขียนโปรแกรมคอมพิวเตอร์ได้

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายเนื้อหา ข้อตกลงในการเรียนการสอน
2. ให้นักศึกษาได้เรียนรู้ผ่านการบรรยายที่จอโปรเจคเตอร์
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 5

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. โปรแกรมคอมพิวเตอร์
4. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
5. แบบฝึกหัดบทที่ 5

การวัดผลและประเมินผล

1. สัมผัสจากการซักถามผู้เรียน
2. สัมผัสจากกิจกรรมร่วม
3. สัมผัสจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด



ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

บทที่ 5

โครงสร้างทางภาษา

หลังจากที่ได้ทดลองเขียนโปรแกรมเพื่อควบคุมปุ่มคอนโทรลต่าง ๆ ในวิชวลเบสิกแล้ว การเขียนโปรแกรมเพื่อควบคุมกระบวนการภายในเช่น บวก ลบ คูณหรือหาร การเก็บค่าจากผู้ใช้ไว้ในตัวแปรต่าง ๆ ฯลฯ จัดเป็นสิ่งจำเป็นอย่างมากเช่นกัน เพราะการเขียนโปรแกรมคอมพิวเตอร์ให้มีประสิทธิภาพไม่ว่าภาษาใด ๆ ก็ตาม จำเป็นต้องศึกษาถึงชนิดของข้อมูลและตัวแปรที่เลือกใช้งาน เนื่องจากข้อมูลต่างชนิดกันใช้เนื้อที่ในหน่วยความจำและวิธีการที่ต่างกันไป การเลือกชนิดข้อมูลให้เหมาะสม การทำความเข้าใจเกี่ยวกับตัวแปร ประเภทและชนิดของตัวแปรแบบต่าง ๆ วิธีการประกาศตัวแปรและการใช้งานตัวแปร ซึ่งภาษาคอมพิวเตอร์แต่ละภาษามีรูปแบบหรือโครงสร้างทางภาษาเป็นของตนเอง และอาจแตกต่างกันบ้างในแต่ละภาษา โครงสร้างทางภาษาจึงเป็นตัวควบคุมให้โปรแกรมทำงานตามข้อกำหนดที่ต้องการ ในการพัฒนาแอปพลิเคชันโดยใช้ภาษาแต่ละภาษา นักศึกษาหรือผู้เขียนโปรแกรมต้องทำความเข้าใจโครงสร้างของภาษาที่ใช้ก่อน เพื่อให้การเขียนโปรแกรมไม่มีข้อผิดพลาด และสามารถทำให้เขียนโปรแกรมได้อย่างมีประสิทธิภาพสูงสุด

ตัวแปร

ตัวแปร (Variable) หมายถึง ชื่อที่ใช้แทนข้อมูลชนิดต่าง ๆ โดยข้อมูลที่จะนำมาเก็บในตัวแปรจะต้องตรงกับชนิดของข้อมูลที่กำหนดไว้ การนำข้อมูลไปใช้งานจะกระทำผ่านตัวแปรที่กำหนดนี้ ซึ่งมีกฎเกี่ยวกับการตั้งชื่อตัวแปรที่สำคัญดังนี้ (ธีรวัฒน์, 2554)

1. ชื่อของตัวแปรต้องเริ่มต้นด้วยตัวอักษร a - z เท่านั้นจะเป็นตัวพิมพ์ใหญ่หรือพิมพ์เล็กก็ได้ ส่วนตัวถัดไปจะเป็นตัวอักษรหรือตัวเลขก็ได้
2. ห้ามใช้อักษรพิเศษอื่น ๆ ที่ไม่ใช่ตัวอักษรหรือตัวเลข เช่น ? , + , - , * , / , ... ยกเว้นเครื่องหมาย_ (underscore) เท่านั้น
3. ต้องไม่ซ้ำกับคำสงวน (Reserved words) ในวิชวลเบสิก

คำสงวน

คำสงวนเป็นคำที่วิชวลเบสิกสงวนไว้ใช้เป็นคำสั่ง ไม่สามารถนำคำเหล่านี้ไปเป็นชื่ออย่างอื่นได้ โดยคำสงวนในวิชวลเบสิกมีดังนี้ (นิรนาม, ม.ป.ป.)

AddHandler	AddressOf	Alias	And
AndAlso	Ansi	As	Assembly
Auto	Boolean	ByRef	Byte
ByVal	Call	Case	Catch
CBool	CByte	CChar	CDate
CDec	CDBl	Char	Clnt
Class	CLng	CObj	Const
CShort	CSng	CStr	CType

Date	Decimal	Declare	Default
Delegate	Dim	DirectCast	Do
Double	Each	Else	Elseif
End	Enum	Erase	Error
Event	Exit	#ExternalSource	False
Finally	For	Friend	Function
Get	GetType	GoTo	Handles
If	Implements	Imports	In
Inherits	Integer	Interface	Is
*Let	Lib	Like	Long
Loop	Me	Mod	Module
MustInherit	MustOverride	MyBase	MyClass
Namespace	New	Next	Not
Nothing	NotInheritable	NotOverridable	Object
On	Option	Optional	Or
Nothing	NotInheritable	NotOverridable	Object
OrElse	Overloads	Overridable	Overrides
ParamArray	Preserve	Private	Property
Protected	Public	RaiseEvent	ReadOnly
ReDim	#Region	REM	RemoveHandler
Resume	Return	Select	Set
Shadows	Shared	Short	Single
Static	Step	Stop	String
Structure	Sub	SyncLock	Then
Throw	To	True	Try
TypeOf	Unicode	Until	*Variant
When	While	With	WithEvents
WriteOnly	Xor	#Const	
#If...Then...#Else	#Region		

ค่าคงที่

ค่าคงที่ (Constant) คือ ค่าที่กำหนดให้กับตัวแปรค่าใดค่าหนึ่ง เพื่อจะนำไปใช้งานตลอดทั้งโปรแกรม โดยไม่สามารถเปลี่ยนค่าตัวแปรนั้นได้ การกำหนดค่าคงที่มีรูปแบบ ดังนี้

Const ชื่อค่าคงที่ = ค่าที่กำหนด เช่น Const PI = 3.141

หรือกำหนดชนิดของข้อมูลด้วย เช่น

Const ชื่อค่าคงที่ As ชนิดของข้อมูล = ค่าที่กำหนด เช่น Const VAT As Integer = 7

ชนิดของข้อมูล

ชนิดของข้อมูล (นัฐพงศ์ , ม.ป.ป.) โดยปกติการกำหนดชื่อค่าคงที่ นิยมใช้ภาษาอังกฤษตัวใหญ่ เพื่อให้แตกต่างจากตัวแปร ซึ่งในตัวอย่างจะพบว่าในการประกาศตัวแปร สิ่งที่จะต้องกระทำคู่กันก็คือ การระบุชนิดของข้อมูลที่จะเก็บในตัวแปรนั้นด้วย

การประกาศตัวแปรและการกำหนดค่าข้อมูลให้กับตัวแปร รูปแบบการประกาศตัวแปรในภาษา Visual Basic จะใช้คำสั่ง Dim ซึ่งย่อมาจากไคน์ (Dimension) โดยมีรูปแบบ ดังนี้

Dim ชื่อตัวแปร As ชนิดของข้อมูล =ค่าที่กำหนดเริ่มต้น

ในการประกาศตัวแปรสิ่งที่จะต้องกระทำคู่กันก็คือการระบุชนิดของข้อมูลที่จะเก็บในตัวแปร นั้นด้วย ซึ่งชนิดของข้อมูลใน Visual Basic มีการแบ่งออกเป็นประเภทและชนิดย่อยต่าง ๆ ดังตารางที่ 5.1 (Loffelmann,& CallaPurohit ,2010)

ตารางที่ 5.1 การระบุประเภทและชนิดของข้อมูล

ประเภท	ชนิด
จำนวนเต็ม	Byte , Short , Integer , Long
ทศนิยม	Single , Double
ตัวอักษร	String , Char
ตรรกะ	Boolean
วันที่	Date/Time
ออบเจกต์	Object

1. ตัวเลขจำนวนเต็ม ข้อมูลประเภทตัวเลขจำนวนเต็มในภาษา Visual Basic สามารถแบ่งย่อย ๆ ได้ดังตารางที่ 5.2

ตารางที่ 5.2 ตัวเลขจำนวนเต็ม

ชนิด	ค่าต่ำสุด	ค่าสูงสุด	ใช้เนื้อที่ใน หน่วยความจำ
Byte	0	255	1 Byte (8 bit)
UShort	0	65,535	2 Byte (16 bit)
UInteger	0	4,294,967,295	4 Byte (32 bit)
ULong	0	18,446,744,073,709,551,615	8 Byte (64 bit)
SByte	-128	127	1 Byte (8 bit)

ชนิด	ค่าต่ำสุด	ค่าสูงสุด	ไซเนื่อที่ใน หน่วยความจำ
Short	-32768	32767	2 Byte (16 bit)
Integer	-2147483648	2147483647	4 Byte (32 bit)
Long	-9223372036854775808	9223372036854775807	8 Byte (64 bit)

2. ตัวเลขทศนิยม เป็นข้อมูลประเภทที่สามารถเก็บตัวเลขประเภททศนิยมได้ โดยสามารถแบ่งออกเป่น 2 ชนิดย่อยดังรายละเอียดในตารางที่ 5.3

ตารางที่ 5.3 ตัวเลขทศนิยม

ชนิด	ค่าต่ำสุด	ค่าสูงสุด	ไซเนื่อที่ใน หน่วยความจำ
Single	-3.402823×10^{38}	3.402823×10^{38}	4 Byte (32 bit)
Double	$-1.79769313486232 \times 10^{308}$	$1.79769313486232 \times 10^{308}$	8 Byte (64 bit)

3. ข้อมูลประเภทตัวเลขที่สามารถเก็บได้ทั้งจำนวนเต็มและตัวเลขทศนิยม โดยมีรายละเอียดดังตารางที่ 5.4

ตารางที่ 5.4 ตัวเลขจำนวนเต็มและตัวเลขทศนิยม

ชนิด	ค่าต่ำสุด	ค่าสูงสุด	ไซเนื่อที่ใน หน่วยความจำ
Decimal	- 79,228,162,514,264,337, 593,543,950,335	79,228,162,514,264,337, 593,543,950,335	8 Byte (64 bit)

4. ตัวอักษร เป็นข้อมูลประเภทที่ตัวอักษรมี 2 ชนิด โดยมีรายละเอียดดังตารางที่ 5.5

ตารางที่ 5.5 ตัวอักษร

ชนิด	ค่าต่ำสุด	ค่าสูงสุด	ไซเนื่อที่ใน หน่วยความจำ
Char	1 ตัวอักษร	1 ตัวอักษร	2 Byte (16 bit)
String	1 ตัวอักษร	เท่ากับความยาวของ ตัวอักษร แต่ไม่เกิน 2MB	แปรผันตาม ความยาว ของ ตัวอักษรแต่ไม่ เกิน 2MB

5. Boolean เป็นชนิดข้อมูลประเภทตรรกะ มีค่า 2 ค่า คือ True กับ False ใช้หน่วยความจำ 2 Byte

6. วันที่/เวลา (Date/time) เป็นข้อมูลที่เก็บข้อมูลวันที่และเวลา ชนิดข้อมูลเป็น Date ใช้เนื้อที่ใน หน่วยความจำ 8 Byte

7. ข้อมูลประเภท Object เป็นชนิดข้อมูลที่ใช้เก็บข้อมูลประเภทพิเศษที่สามารถใช้แทนชนิดอื่น ๆ ได้ โดยค่าเริ่มต้นเป็น Nothing และใช้หน่วยความจำ 4 Byte

8. ข้อมูลประเภท Array เป็นข้อมูลที่เก็บข้อมูลอะเรย์

9. User Define Type / Structure เป็นชนิดข้อมูลที่เราร่างขึ้นมาใช้งานเองได้ เช่น ถ้าเราต้องการสร้างตัวแปรเพื่อเก็บข้อมูลลูกค้า เกี่ยวกับ ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ ฯลฯ ตัวอย่าง

```
Dim Name As String
Dim Address As String
Dim Tel As Integer
Dim ID As String
Dim Age As Byte
```

การกำหนดค่าให้กับตัวแปร

การกำหนดค่า (Assignment) ให้กับตัวแปร สำหรับชนิดของข้อมูลแต่ละชนิดจะมีรูปแบบแตกต่างกัน โดยใช้เครื่องหมาย “ = ” ในการให้ค่ากับตัวแปร

1. ข้อมูลชนิดตัวเลข เช่น Byte, Integer, Double จะสามารถใส่ตัวเลขลงไปได้เลย (ไม่ต้องมีเครื่องหมาย ,) เช่น

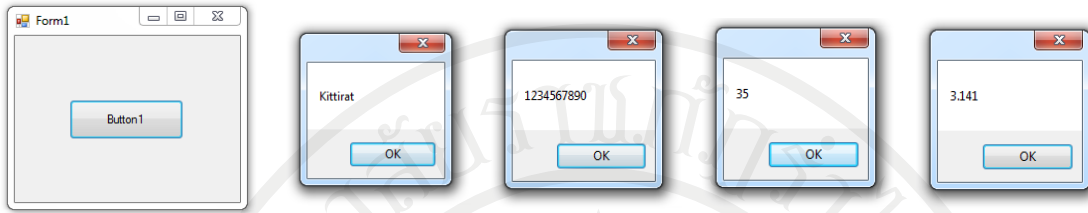
```
Dim x As Integer = 4851
Dim y As Double = 1422.576
```

2. ข้อมูลชนิดบูลีน (Boolean) จะมีค่าเป็นไปได้เพียง 2 ค่า คือ True หรือ False อย่างไม่อย่างหนึ่ง สามารถนำไปกำหนดค่าให้กับตัวแปรได้ เช่น

```
Dim a As Boolean = True
Dim b As Boolean = False
```

3. ข้อมูลชนิดสตริง (String) คือ ข้อมูลที่เป็นสายอักขระหรือเป็นการนำเอาอักขระแต่ละตัวมาวางเรียงต่อกัน ซึ่งข้อมูลชนิดสตริงจะมีความยาวเท่าไรก็ได้ และไม่จำเป็นจะต้องเขียนอักขระทุกตัวติดกัน การกำหนดข้อมูลที่เป็นสตริง จะต้องกำหนดจุดเริ่มต้นและสิ้นสุดของสตริงเสมอ โดยใช้เครื่องหมาย “ ”

ตัวอย่างที่ 5.1 ทดลอง ชนิดของข้อมูล ตัวแปรและค่าคงที่
ให้นักศึกษาทดลองป้อนคำสั่งโปรแกรมต่อไปนี้เพื่อให้ได้ผลลัพธ์ดังภาพที่ 5.1



ภาพที่ 5.1 ทดลอง ชนิดของข้อมูล ตัวแปรและค่าคงที่
โปรแกรมคำสั่ง

Public Class Form1

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim Name As String = "Kittirat"

Dim Tel As Integer = 1234567890

Dim Age As Byte = 35

Const PI = 3.141

MessageBox.Show(Name)

MessageBox.Show(Tel)

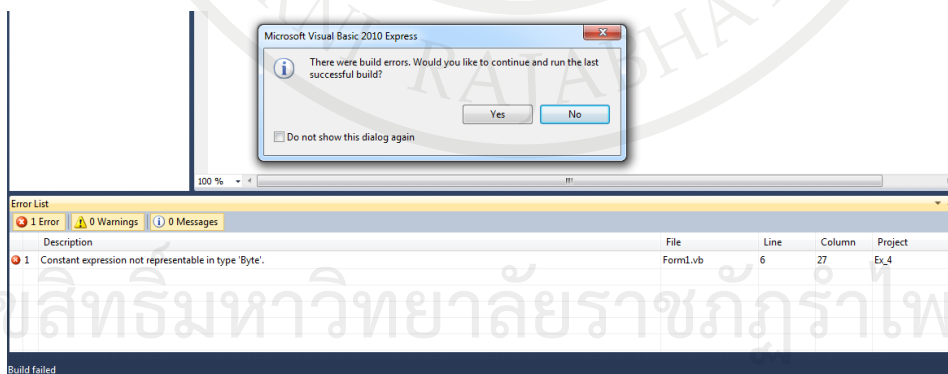
MessageBox.Show(Age)

MessageBox.Show(PI)

End Sub

End Class

ในกรณีที่มีการใส่ข้อมูลเกินหรือผิดจากชนิดของข้อมูลเช่น จาก Dim Age As Byte = 35 ถ้าเปลี่ยนเป็น Dim Age As Byte = 3500 ซึ่ง Byte สามารถใส่ค่าสูงสุดได้ 255 จะเกิดกล่องข้อความเตือนดังภาพที่ 5.2



ภาพที่ 5.2 การแจ้งเตือนการใส่ข้อมูลเกินหรือผิดจากชนิดของข้อมูล

โอเปอเรเตอร์

โอเปอเรเตอร์ (Operator) คือ สัญลักษณ์ที่ใช้ในการประมวลผลทางคณิตศาสตร์หรือทางตรรกะ ภาษาวิซวลเบสิกแบ่งโอเปอเรเตอร์ได้ดังนี้

1. โอเปอเรเตอร์ทางคณิตศาสตร์ เป็นเครื่องหมายสำหรับการคำนวณทางคณิตศาสตร์ ดังตารางที่ 5.6

ตารางที่ 5.6 โอเปอเรเตอร์ทางคณิตศาสตร์

Operator	ความหมาย
+	บวก
-	ลบ
*	คูณ
/	หาร
^	ยกกำลัง เช่น $2^3 = 8$
Mod	หารเอาเฉพาะเศษ $8 \text{ Mod } 3 = 2$
\	หารโดยเอาเฉพาะจำนวนเต็ม ตัดเศษทิ้ง เช่น $10 \setminus 3 = 3$

2. โอเปอเรเตอร์สำหรับการเปรียบเทียบ เพื่อหาค่าความจริงระหว่าง 2 นิพจน์ ซึ่งหากผลการเปรียบเทียบเป็นจริง จะได้ผลลัพธ์เป็น True แต่หากผลการเปรียบเทียบเป็นเท็จจะได้ผลลัพธ์เป็น False ดังตารางที่ 5.7

ตารางที่ 5.7 โอเปอเรเตอร์สำหรับการเปรียบเทียบ

Operator	ความหมาย
=	ใช้ในการเปรียบเทียบว่าเท่ากันหรือไม่ $y = 10, z = 10, c = (y = z)$
< >	ไม่เท่ากับ $y = 10, z = 10, c = (y <> z)$
>	มากกว่า $c = (10 > 5)$
<	น้อยกว่า $c = (10 < 5)$
> =	มากกว่าหรือเท่ากับ $c = (10 >= 5)$
< =	น้อยกว่าหรือเท่ากับ $c = (10 <= 5)$

3. โอเปอเรเตอร์สำหรับการกำหนดค่า เป็นเครื่องหมายสำหรับการกำหนดค่าของตัวแปรทางด้านซ้ายของเครื่องหมายด้วยค่าที่อยู่ทางด้านขวาของเครื่องหมายดังตารางที่ 5.8

ตารางที่ 5.8 โอเปอเรเตอร์สำหรับการกำหนดค่า

Operator	ความหมาย
=	เท่ากับ เช่น $X = 10$
^ =	ยกกำลังด้วย เช่น $2^3 = 8$ หมายถึง 2 ยกกำลังด้วย 3 เท่ากับ 8
* =	คูณด้วย เช่น $X * = 10$ หมายถึง $X = X * 10$

Operator	ความหมาย
/ =	หารด้วย เช่น $X / = 10$ หมายถึง $X = X / 10$
\ =	หารด้วย เช่น $X \ = 10$ หมายถึง $X = X \ 10$
+ =	บวกด้วย เช่น $X += 10$ หมายถึง $X = X + 10$
- =	ลบด้วย เช่น $X -= 10$ หมายถึง $X = X - 10$
	ใช้สำหรับข้อมูลชนิดสตริงเพื่อนำข้อมูลชนิดชนิดสตริงมาต่อท้าย
& =	$x = \text{"Hello"} \ y = \text{"World"} \ z = x \ \& \ y$
	$x \ \&= \ y$ จะได้ $x = \text{"Hello World"}$

4. โอเปอเรเตอร์ทางตรรกะ (Logical) เป็นการเปรียบเทียบระหว่าง 2 นิพจน์ ซึ่งแต่ละนิพจน์ที่นำมาเปรียบเทียบจะต้องมีค่าเป็น True หรือ False ใดๆอย่างหนึ่ง และผลลัพธ์ที่ได้ก็จะเป็น True หรือ False ใดๆอย่างหนึ่งด้วยเช่นกัน ดังตารางที่ 5.9

ตารางที่ 5.9 โอเปอเรเตอร์ทางตรรกะ

Operator	ความหมาย
And	ถ้าทั้งสองนิพจน์เป็นจริงทั้งคู่ ผลที่ได้จะเป็นจริง นอกนั้นเป็นเท็จหมด
Or	ถ้าทั้งสองนิพจน์เป็นเท็จทั้งคู่ ผลที่ได้จะเป็นเท็จ นอกนั้นเป็นจริงหมด
Not	ผลที่ได้จะเป็นค่าตรงกันข้ามกับค่าของนิพจน์เดิม
Xor	ถ้านิพจน์แรกกับนิพจน์หลังเหมือนกัน (จริง - จริง / เท็จ - เท็จ) ผลที่ได้จะเป็นเท็จ นอกนั้นเป็นจริง

5. โอเปอเรเตอร์สำหรับการเชื่อมต่อสตริง (String Concatenation) การเชื่อมต่อข้อความหรือสตริงเข้าด้วยกัน จะใช้เครื่องหมายอยู่ 2 ตัว คือเครื่องหมาย “+” และเครื่องหมาย “&” ถ้าเป็นการเชื่อมต่อสตริงกับสตริง เข้าด้วยกัน สามารถใช้เครื่องหมาย + หรือ เครื่องหมาย & ก็ได้ เช่น

“Hello” + “World” หรือ “Hello” & “World”

แต่ถ้าเป็นการเชื่อมต่อระหว่างสตริงกับตัวเลข จะต้องใช้เครื่องหมาย & เช่น

“Visual Basic” & 2010

ถ้าต้องการเครื่องหมาย + เพื่อเชื่อมต่อสตริงกับตัวเลขจะต้องแปลงตัวเลขให้อยู่ในรูปแบบสตริงก่อน ซึ่งมีวิธีการ ดังนี้

- ใช้ฟังก์ชัน CStr() เช่น “Visual Basic” + CStr(2010)
- ใช้ฟังก์ชัน ToString() เช่น “Visual Basic” + 2010.ToString()

ลำดับในการทำงานของโอเปอเรเตอร์

ลำดับการทำงานก่อนหลังของโอเปอเรเตอร์ชนิดต่าง ๆ ดังตารางที่ 5.10 จะเรียงลำดับการทำงานจากสูงสุดไปยังต่ำสุด การคำนวณค่าของนิพจน์ใด ๆ ให้ดูว่าโอเปอเรเตอร์ใดมีลำดับในการทำงานสูงสุดก็ให้หาค่าจากการกระทำของโอเปอเรเตอร์นั้นก่อน ตามด้วยหาค่าจากการกระทำของโอเปอเรเตอร์ที่มีลำดับในการทำงานต่ำลงมานิพจน์บางรูปแบบจะมีโอเปอเรเตอร์ในนิพจน์ที่มีลำดับการทำงานผลลัพธ์เท่ากัน เราจะหาค่าการกระทำของโอเปอเรเตอร์นั้นจากซ้ายไปขวา

ตารางที่ 5.10 ลำดับในการทำงานของโอเปอเรเตอร์

ชนิดของโอเปอเรเตอร์	โอเปอเรเตอร์	คำอธิบาย
โอเปอเรเตอร์คำนวณ	\wedge	การยกกำลัง
	-	การบอกว่าเป็นจำนวนลบ
	*, /	การคูณ, การหาร
	\	การหารจำนวนเต็ม
	Mod	การหาเศษจากการหาร
	+, -	การบวก, การลบ
โอเปอเรเตอร์เชื่อมต่อสตริง	+, &	การเชื่อมต่อสตริง
โอเปอเรเตอร์เปรียบเทียบ	=, <>, >=, <=, <, >, Like, Is	การเปรียบเทียบ
โอเปอเรเตอร์ตรรกะ	Not	
	And, AndAlso	
	Or, OrElse	
	Xor	

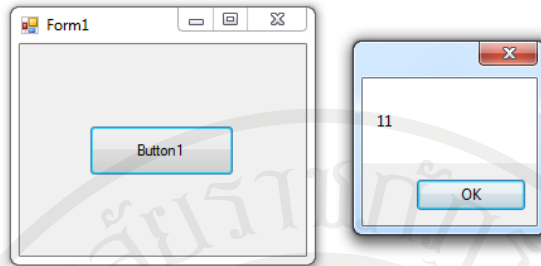
ตัวอย่างที่ 5.2 ทดลองโอเปอเรเตอร์ทางคณิตศาสตร์

ให้นักศึกษาทดลองป้อนคำสั่งโปรแกรมต่อไปนี้เพื่อทดลองโอเปอเรเตอร์ทางคณิตศาสตร์ โดยให้ได้ผลลัพธ์ดังภาพที่ 5.3

โปรแกรมคำสั่ง

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
    Dim num As Byte
    num = 5 + 3 * 2
    MessageBox.Show(num)
End Sub
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 5.3



ภาพที่ 5.3 ผลลัพธ์โอเพอร์เรเตอร์ทางคณิตศาสตร์

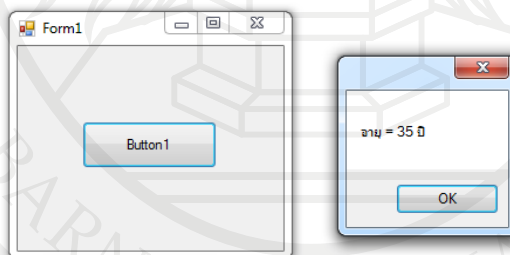
ตัวอย่างที่ 5.3 โอเพอร์เรเตอร์สำหรับการเชื่อมต่อสตริง

ให้นักศึกษาทดลองป้อนคำสั่งโปรแกรมต่อไปนี้เพื่อทดลองโอเพอร์เรเตอร์ทางคณิตศาสตร์ โดยให้ได้ผลลัพธ์ดังภาพที่ 5.4

โปรแกรมคำสั่ง

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Age As Byte = 35
    MessageBox.Show("อายุ = " & Age.ToString & " ปี")
End Sub
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 5.4



ภาพที่ 5.4 โอเพอร์เรเตอร์สำหรับการเชื่อมต่อสตริง

การเขียนคำอธิบายโปรแกรม

การเขียนคำอธิบายโปรแกรม (Comment) คือ ข้อความที่ผู้พัฒนาโปรแกรมต้องการที่จะบรรยายคำสั่ง หรือเป็นสิ่งเตือนความจำ เมื่อทำการแก้ไขเปลี่ยนแปลงโค้ดโปรแกรม ก็สามารถศึกษาได้จากคำอธิบายโปรแกรม ซึ่งคำอธิบายโปรแกรมหากล่าว โปรแกรมจะไม่นำมาประมวลผลด้วย

ในการเขียนคำอธิบายในวิซวลเบสิกจะใช้เครื่องหมาย “ ‘ ” วางไว้หน้าคำอธิบายโปรแกรมนั้นตั้งแต่เครื่องหมาย ‘ ไปจนจบบรรทัดโปรแกรมจะถือว่าเป็นคำอธิบายโปรแกรม เช่น

Dim x As Integer 'x เป็นตัวแปรชนิดจำนวนเต็ม

ข้อสังเกต: คำอธิบายโปรแกรมจะแสดงข้อความเป็นสีเขียว

คำสั่งในการเขียนโปรแกรม

คำสั่ง (Statement) คือ คำสั่งที่สั่งให้คอมพิวเตอร์ทำงานโดวิซวลเบสิกจะถือว่า 1 บรรทัด คือ 1 คำสั่งหรือ เมื่อสิ้นสุดบรรทัด ก็ถือว่าเป็นการสิ้นสุดคำสั่ง เช่น

Dim a As Integer
a += 10

ในกรณีที่ต้องการเขียนคำสั่งไว้ในบรรทัดเดียวกันหลาย ๆ คำสั่งให้ใช้เครื่องหมาย “ : ” คั่นระหว่างคำสั่งเช่น

Dim a, b, c As Integer
a = 30 : b = 40 : c = 50

ในกรณีที่คำสั่งมีความยาวมากจำเป็นจะต้องพิมพ์หลายบรรทัดให้ใช้เครื่องหมาย “ _ ” วางไว้ท้ายบรรทัดที่ต้องการให้ขึ้นบรรทัดใหม่ เช่น

Dim a As _
Integer = 20

การรับข้อมูลและแสดงข้อความ

การรับข้อมูลและแสดงข้อความ

1. การรับข้อมูลเป็นการให้ผู้ใช้สามารถกำหนดข้อมูลลงไปเองตามต้องการ โดยวิซวลเบสิกจะใช้ฟังก์ชัน InputBox() ในการรับข้อมูลจากผู้ใช้ เพื่อนำไปประมวลผลรูปแบบ

InputBox (“ข้อความ”)

- เนื่องจากฟังก์ชันนี้ใช้รับข้อมูล ดังนั้นจะต้องสร้างตัวแปรไปรับค่าจาก InputBox() เช่น

Dim name As String
name = InputBox (“กรุณาป้อนชื่อ - นามสกุล”)
name = InputBox (“กรุณาป้อนชื่อ - นามสกุล”, “ป้อนข้อมูล”)

2. การขึ้นบรรทัดใหม่ ปกติแล้วสตริงทั้งหมดจะถูกเขียนที่บรรทัดเดียวกัน ซึ่งบางครั้งอาจมีความยาวมากเกินไป ดังนั้นวิซวลเบสิกจะมีคำสั่งที่ใช้กำหนดให้ขึ้นบรรทัดใหม่ คือ vbNewLine โดยการนำมาเชื่อมต่อกับสตริงด้วยโอเปอร์เรเตอร์ “&” ณ จุดที่ต้องการขึ้นบรรทัดใหม่ เช่น

```
Dim str As String
str = "Visual Studio 2010 ประกอบด้วย" & vbNewLine
str &= "Visual Basic 2010" & vbNewLine
str &= "Visual C# 2010 " & vbNewLine
str &= "Visual C++ 2010"
MessageBox.Show (str, "แสดงข้อความขึ้นบรรทัดใหม่")
```

สรุป

การเขียนโปรแกรมคอมพิวเตอร์ใหม่มีประสิทธิภาพไม่ว่าภาษาใด ๆ ก็ตามจำเป็นต้องศึกษาถึงตัวแปร ที่เป็นชื่อสำหรับใช้แทนข้อมูลชนิดต่าง ๆ ที่ไม่ตรงกับคำสั่ง โดยตัวแปรนี้อาจมีการกำหนดค่าคงที่ต่าง ๆ เพื่อนำไปใช้ในโปรแกรม และต้องสัมพันธ์กับการกำหนดชนิดของข้อมูล ในแต่ละประเภท เช่น จำนวนเต็ม, ทศนิยม, ตัวอักษร, ตรรกะ, วันที่, หรือออบเจกต์ ต่าง ๆ ทั้งนี้ การศึกษาโอเปอร์เรเตอร์ ที่เป็นสัญลักษณ์ในการประมวลผลทางคณิตศาสตร์หรือทางตรรกะ นั้นมีลำดับก่อนหลังที่หากศึกษาไม่เข้าใจจะส่งผลถึงการคำนวณของโปรแกรมคอมพิวเตอร์ที่ผิดพลาดได้ อีกทั้งโครงสร้างทางภาษาของวิซวลเบสิกนั้นมีวิธีเขียนเฉพาะเช่น ใช้เครื่องหมาย “ ” วางไว้หน้าคำอธิบายโปรแกรม เพื่อบอกคำอธิบายโปรแกรม 1 บรรทัด คือ 1 คำสั่งหากต้องการเขียนคำสั่ง ไว้ในบรรทัดเดียวกันหลาย ๆ คำสั่งให้ใช้เครื่องหมาย “ : ” คั่นระหว่างคำสั่งหรือ ในกรณีที่คำสั่งมีความยาวมากให้ใช้เครื่องหมาย “ _ ” วางไว้ท้ายบรรทัดที่ต้องการให้ขึ้นบรรทัดใหม่ เป็นต้น ซึ่งเมื่อนักศึกษาหรือผู้เขียนโปรแกรมได้ทำความเข้าใจโครงสร้างของภาษา จะช่วยให้การเขียนโปรแกรมไม่มีข้อผิดพลาดและสามารถทำให้เขียนโปรแกรมได้อย่างมีประสิทธิภาพสูงสุด

แบบฝึกหัดบทที่ 5

1. ข้อใดไม่สามารถใช้เขียนเพื่อกำหนดเป็นชื่อตัวแปรได้
 - 1.1) Reserved
 - 1.2) A123
 - 1.3) 1_Label
 - 1.4) Option
 - 1.5) 12345
2. การเขียนโปรแกรมเพื่อกำหนดค่าคงที่มีรูปแบบการเขียนอย่างไร
3. จงยกตัวอย่างวิธีการเขียน ตัวแปร ค่าคงที่และชนิดของข้อมูลมา 3 ตัวอย่าง
4. ชนิดของข้อมูลแบบ Integer ใช้เนื้อที่ในหน่วยความจำ ค่าสูงสุดและต่ำสุดเท่าไร
5. ชนิดของข้อมูล Char และ String คืออะไร เหมือนหรือต่างกันอย่างไร อธิบายพร้อมยกตัวอย่าง
6. ตัวแปร A มีค่าเท่าไรหากป้อนสูตรคำนวณว่า $A = 5 + 2 \times 3$
7. โอเปอเรเตอร์ <> คืออะไร อธิบายพร้อมยกตัวอย่างการใช้งาน
8. หากต้องการเชื่อมต่อข้อความหรือสตริงเข้าด้วยกันต้องทำอย่างไร อธิบายพร้อมยกตัวอย่าง
9. ระหว่างยกกำลัง บวก ทหาร คูณ คอมพิวเตอร์มีลำดับในการทำงานของโอเปอเรเตอร์ใดก่อน
10. การใช้ “ ‘ ”, “ : ” และ “ _ ” ในการเขียนโปรแกรมจะเกิดผลอย่างไรบ้างอธิบายพร้อมยกตัวอย่าง

เอกสารอ้างอิง

- นัฐพงศ์ สงเนียม, (ม.ป.ป.). การเขียนโปรแกรมด้วย Microsoft Visual Studio 2010. [On-line] Available : http://www.siam2dev.com/e_learning/algorithm/lec04_vb_net2010_variables_declaration.pdf. [2556, December 8].
- นิรนาม, (ม.ป.ป.). Visual basic 2010 gramming. [On-line] Available : <http://visualgramming.blogspot.com/2013/11/reserved-words.html#> [2556, December 8].
- ธีรวัฒน์ ประกอบผล, (2554). คู่มือการเขียนแอปพลิเคชันด้วย Visual Basic 2010. กรุงเทพมหานคร: ชิมพลีฟาย.
- สัจจะ จรัสรุ่งรวีวร, (2554). คู่มือเรียนและใช้งาน Visual Basic 2010. นนทบุรี: ไอดีซี พรีเมียร์จำกัด.
- Loffelmann, K., & CallaPurohit ,S.(2010). Microsoft Visual Basic 2010 Developer's Handbook, Canada: O' Reilly Media.

แผนบริหารการสอนประจำบทที่ 6

หัวข้อเนื้อหาบทที่ 6

1. คำสั่งควบคุมการทำงานแบบลำดับ
2. คำสั่งควบคุมการทำงานแบบเลือกเส้นทาง
3. คำสั่งควบคุมการทำงานแบบทำซ้ำ
4. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 6 แล้วนักศึกษาสามารถ

1. สามารถเขียนแอฟพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์โดยการเขียนคำสั่งควบคุมการทำงานแบบลำดับได้
2. สามารถเขียนแอฟพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์โดยการเขียนคำสั่งควบคุมการทำงานแบบเลือกเส้นทางได้
3. สามารถเขียนแอฟพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์โดยการเขียนคำสั่งควบคุมการทำงานแบบทำซ้ำได้

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายเนื้อหา
2. ให้นักศึกษาได้เรียนรู้ผ่านการบรรยายที่จอโปรเจคเตอร์
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 6

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. โปรแกรมคอมพิวเตอร์
4. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
5. แบบฝึกหัดบทที่ 6

การวัดผลและประเมินผล

1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากกิจกรรมร่วม
3. สังเกตจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด

บทที่ 6

การเขียนแอปพลิเคชันเบื้องต้นเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์

จากการศึกษาการเขียนผังงาน การออกแบบหน้าจอ โครงสร้างทางภาษากำหนดตัวแปร เบื้องต้นและได้ทดลองทำแบบฝึกหัดต่าง ๆ จนเกิดความรู้ความสามารถแล้ว การทดลองเขียนโปรแกรมเพื่อประมวลผลข้อมูลด้วยคอมพิวเตอร์ในบทนี้จะเป็นการเขียนโปรแกรมตามสภาพปัญหาที่อาจเกิดขึ้นทั่วไปโดยมีความซับซ้อนที่ต่างกัน ซึ่งมีทั้งการเขียนโปรแกรมแบบเป็นลำดับลงมาเรื่อย ๆ หรือการเขียนโปรแกรมที่สามารถควบคุมทิศทางการทำงานได้ เช่น ข้อมูลที่รับเข้ามาเป็นเลขคู่ ให้ทำคำสั่งปฏิบัติงานอย่างหนึ่งหรืออาจให้ทำคำสั่งรับค่ามาจนถึงค่า ๆ หนึ่งแล้ว ให้โปรแกรมทำงานอีกแบบหนึ่ง เป็นต้นตามความซับซ้อนของสภาพงานที่ผู้ใช้ต้องการ โดรนเนื้อหาในบทนี้จะเป็นการรวมความรู้ที่เคยศึกษาในบทก่อน ๆ มาประยุกต์ใช้อย่างการเขียนผังงานมาใช้ในการเขียนแอปพลิเคชัน ตลอดจนมีวิธีการเขียนมากมายให้นักศึกษาได้เรียนรู้และใช้งาน

คำสั่งควบคุมการทำงานแบบลำดับ

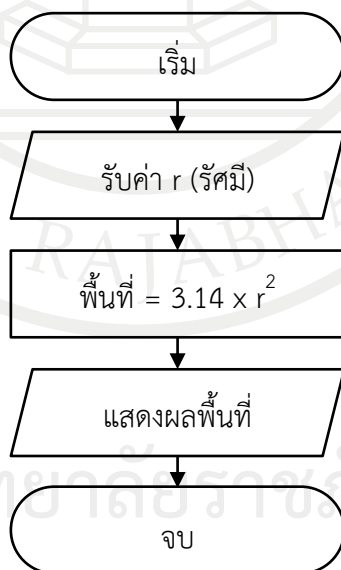
โดยโครงสร้างควบคุมการทำงานที่ละคำสั่งจากบนลงล่าง รวมถึงการรับข้อมูลเข้า และการเอาข้อมูลแสดงผลออก โดยศึกษาได้ตามตัวอย่างต่อไปนี้

ตัวอย่างที่ 6.1 คำนวณหาพื้นที่วงกลม

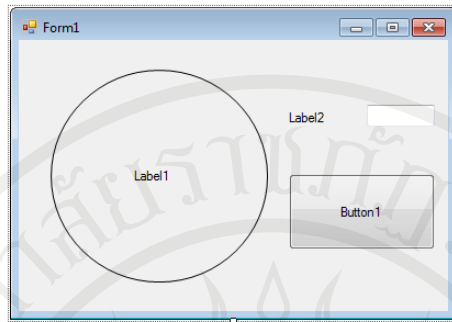
อัลกอริทึม

1. รับค่า r (รัศมี)
2. คำนวณตามสูตร $\text{พื้นที่} = 3.14 \times r^2$
3. แสดงผลพื้นที่

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.1



ภาพที่ 6.1 โปรแกรมคำนวณหาพื้นที่วงกลม

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles MyBase.Load
```

```
        Label1.Text = "พื้นที่วงกลม"
```

```
        Label2.Text = "รัศมีวงกลม = "
```

```
        Button1.Text = "คำนวณ"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles Button1.Click
```

```
        Dim area As Single
```

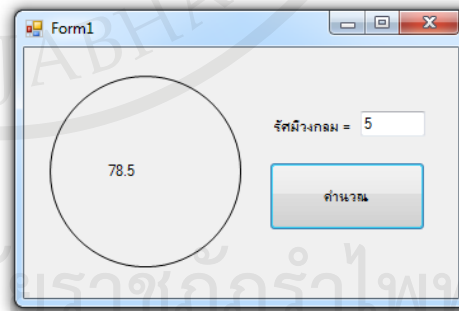
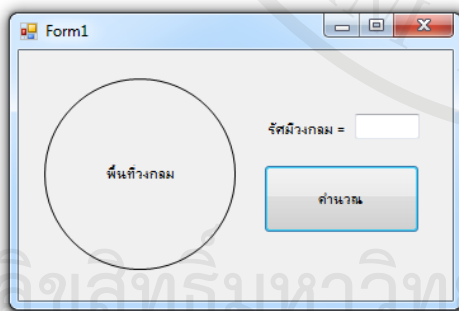
```
        area = 3.14 * TextBox1.Text ^ 2
```

```
        Label1.Text = area
```

```
    End Sub
```

```
End Class
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.2



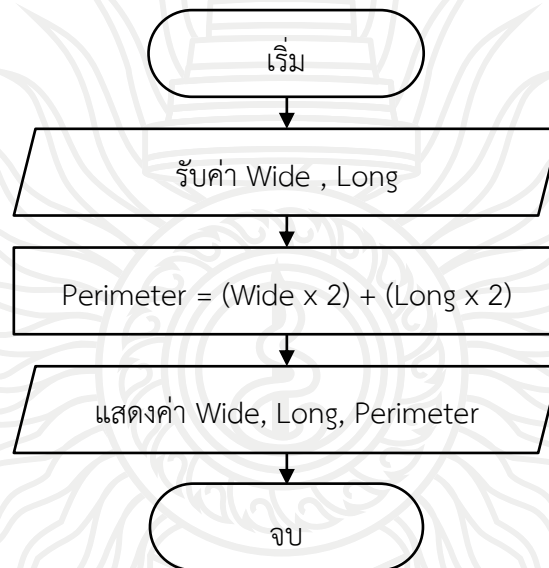
ภาพที่ 6.2 ผลลัพธ์โปรแกรมคำนวณหาพื้นที่วงกลม

ตัวอย่างที่ 6.2 คำนวณหาเส้นรอบรูปสี่เหลี่ยมผืนผ้า

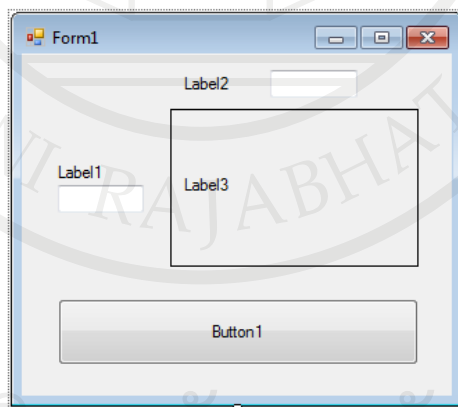
อัลกอริทึม

1. เริ่มต้นการทำงาน
2. รับค่า ความกว้าง Wide , ความยาว Long
3. คำนวณพื้นที่สี่เหลี่ยมผืนผ้าจากสูตร $Perimeter = (Wide \times 2) + (Long \times 2)$
4. แสดงค่า Wide , Long , Perimeter
5. จบการทำงาน

ผังงาน



ออกแบบหน้าจอจดังภาพที่ 6.3



ภาพที่ 6.3 โปรแกรมคำนวณหาเส้นรอบรูปสี่เหลี่ยมผืนผ้า

โปรแกรมคำสั่ง

Public Class Form1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label1.Text = " ความกว้าง "

Label2.Text = " ความยาว "

Label3.Text = " ความยาวเส้นรอบรูปสี่เหลี่ยมผืนผ้า "

Button1.Text = " คำนวณ "

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim Perimeter As Single

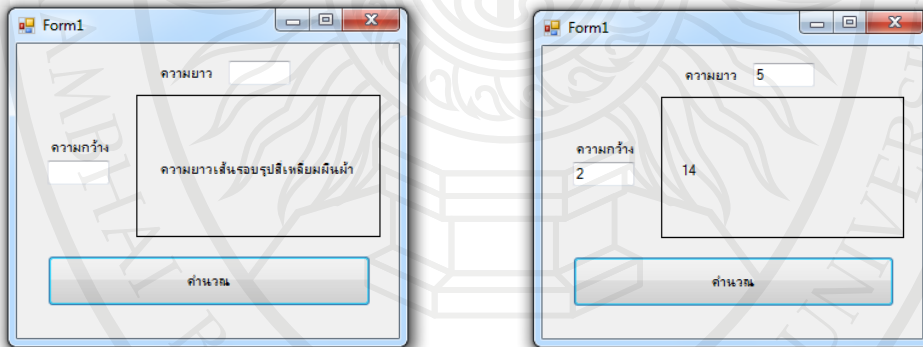
Perimeter = (TextBox1.Text * 2) + (TextBox2.Text * 2)

Label3.Text = Perimeter

End Sub

End Class

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.4



ภาพที่ 6.4 ผลลัพธ์โปรแกรมคำนวณหาเส้นรอบรูปสี่เหลี่ยมผืนผ้า

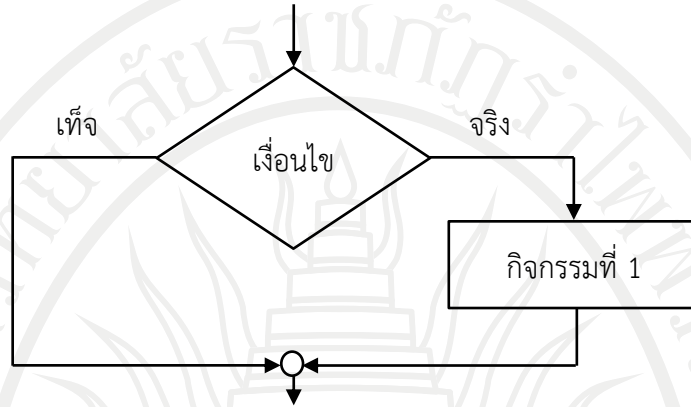
คำสั่งควบคุมการทำงานแบบเลือกเส้นทาง

เป็นโครงสร้างการทำงานที่มีการตัดสินใจโดยมีเงื่อนไขเป็นตัวกำหนด เป็นกลุ่มคำสั่งที่ต้องใช้มากที่สุดในการเขียนโปรแกรม ไม่ว่าจะเขียนเป็นภาษาใดก็ตาม โดยกลุ่มคำสั่งที่เกี่ยวกับการสร้างเงื่อนไขแบ่งได้เป็น 2 คำสั่งใหญ่ ๆ คือ

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

1. คำสั่ง If ... Then สามารถแบ่งได้ 3 แบบ

1.1 คำสั่ง If ... Then แบบที่ 1 เป็นการตรวจสอบเงื่อนไขที่เราสนใจ ถ้าเป็นจริงให้ทำตามรูปแบบนั้น



ภาพที่ 6.5 ผังงานแบบเลือกเส้นทางด้วยคำสั่ง If ... Then แบบที่ 1
จากภาพที่ 6.5 สามารถ เขียนคำสั่ง If ... Then ได้ดังนี้

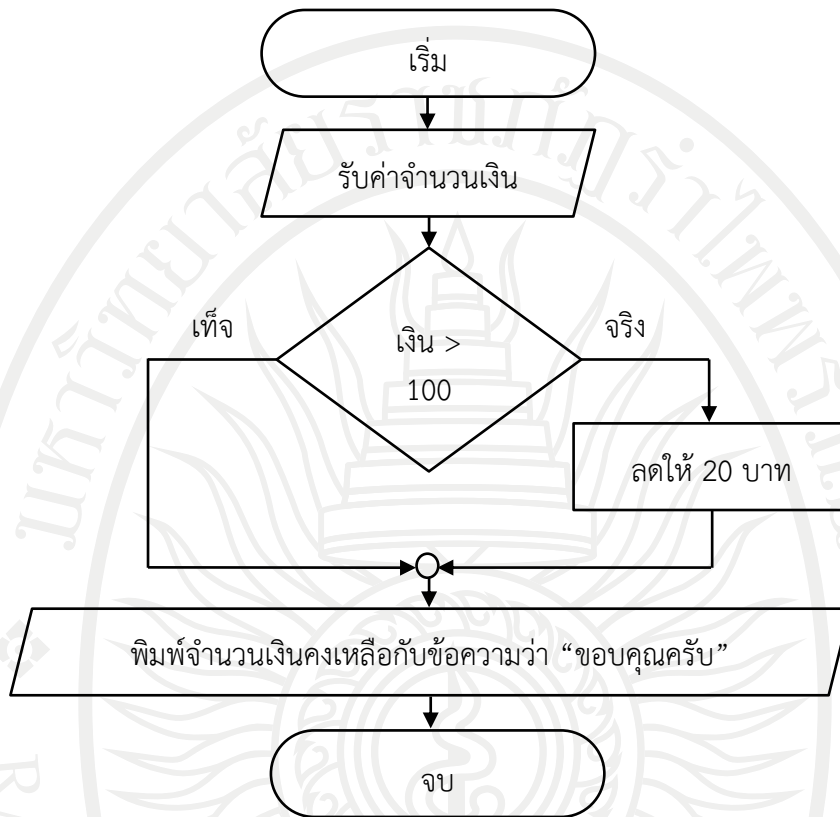
If เงื่อนไข Then กิจกรรมที่ 1

ตัวอย่าง 6.3 การลดราคาเมื่อสินค้าเกิน 100 บาท

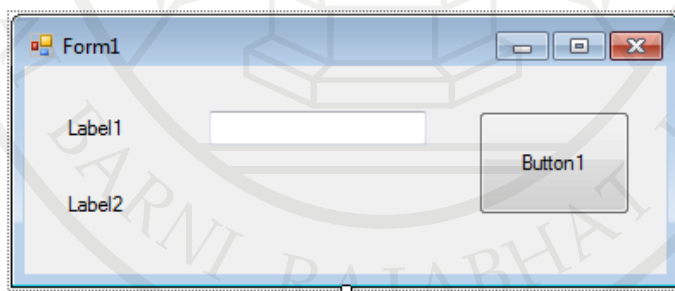
อัลกอริทึม

1. รับค่าเงิน
2. ตรวจสอบเงื่อนไขว่าถ้าเกิน 100 บาท
เงื่อนไขเป็นจริงให้ ลดราคา 10 ให้ 20 บาท
3. พิมพ์ จำนวนเงินคงเหลือกับข้อความว่า “ขอบคุณครับ”

ผังงาน



ออกแบบหน้าจอ ดังภาพที่ 6.6



ภาพที่ 6.6 หน้าจอโปรแกรมการลดราคาเมื่อสินค้าเกิน 100 บาท

โปรแกรมคำสั่ง

Public Class Form1

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load

```

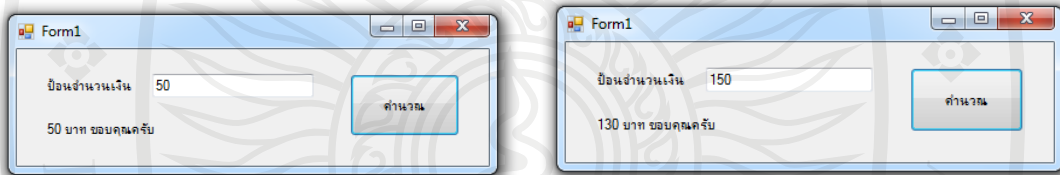


```

Label1.Text = "ป้อนจำนวนเงิน"
Label2.Text = " "
Button1.Text = "คำนวณ"
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
    Dim money As Integer
    money = TextBox1.Text
    If money > 100 Then money = money - 20
    Label2.Text = money & " บาท ขอขอบคุณครับ"
End Sub
End Class

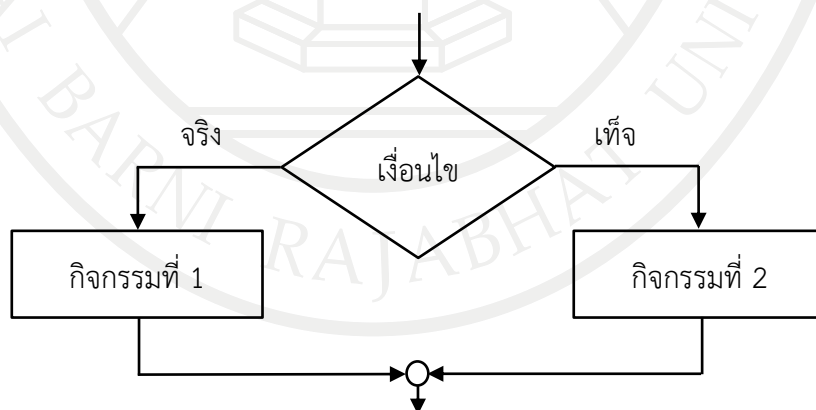
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.7



ภาพที่ 6.7 ผลลัพธ์โปรแกรมการลดราคาเมื่อสินค้าเกิน 100 บาท

1.2 คำสั่ง If ... Then แบบที่ 2 เป็นคำสั่งในการเลือกทำคำสั่งใดคำสั่งหนึ่งเท่านั้น โดยตรวจสอบเงื่อนไขว่าเป็นจริงหรือเท็จ



ภาพที่ 6.8 ผังงานแบบเลือกเส้นทางด้วยคำสั่ง If ... Then แบบที่ 2

จากภาพที่ 6.8 สามารถเขียนคำสั่ง If ... Then ได้ดังนี้

```

If เงื่อนไข Then
    กิจกรรมที่ 1
Else
    กิจกรรมที่ 2
End If
  
```

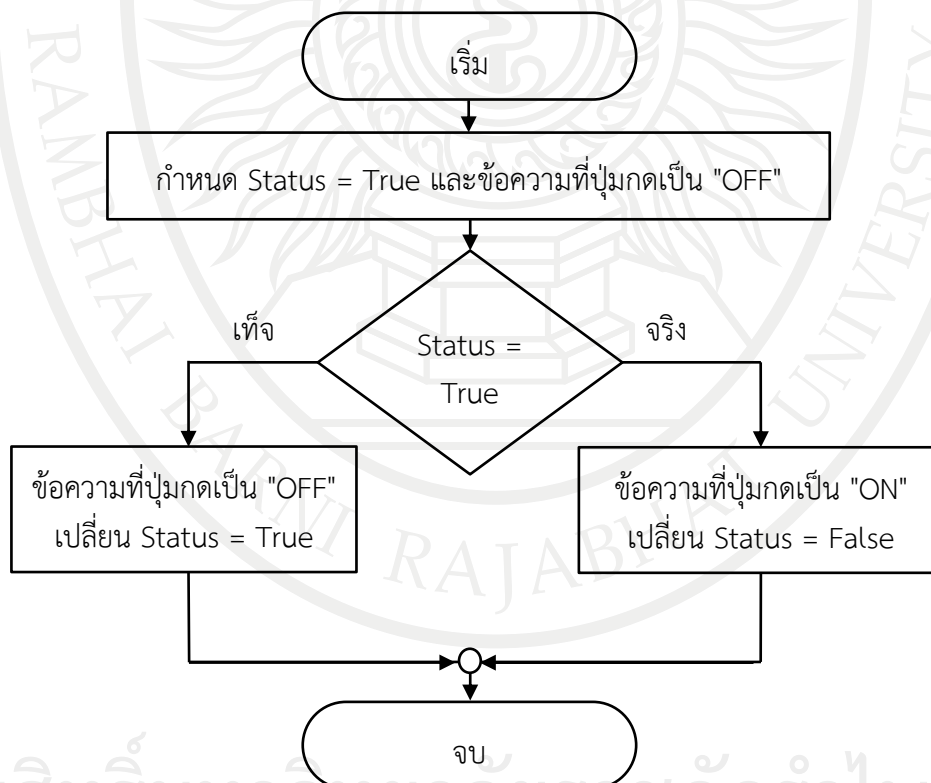
หมายเหตุ Else หมายถึงเงื่อนไขเป็นเท็จ

ตัวอย่าง 6.4 การทำปุ่มสำหรับกด

อัลกอริทึม

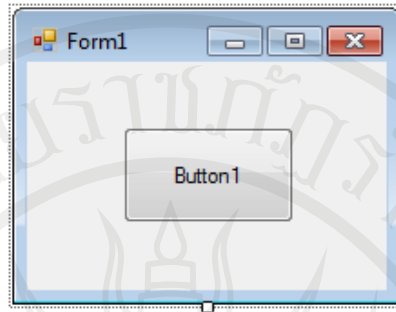
1. กำหนดสถานะเริ่มต้นให้ Status = True และข้อความที่ปุ่มกดเป็น "OFF"
2. ตรวจสอบเงื่อนไขว่า Status = True
 เงื่อนไขเป็นจริงให้ ข้อความที่ปุ่มกดเป็น "ON" และเปลี่ยน Status = False
 เงื่อนไขเป็นเท็จให้ ข้อความที่ปุ่มกดเป็น "OFF" และเปลี่ยน Status = True

ผังงาน



ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ออกแบบหน้าจอตั้งภาพที่ 6.9



ภาพที่ 6.9 หน้าจอโปรแกรมปุ่มสำหรับกด

โปรแกรมคำสั่ง

```
Public Class Form1
    Dim Status As Boolean
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load
        Button1.Text = "OFF"
        Status = True
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
        If Status = True Then
            Button1.Text = "ON"
            Status = False
        Else
            Button1.Text = "OFF"
            Status = True
        End If
    End Sub
End Class
```

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.10



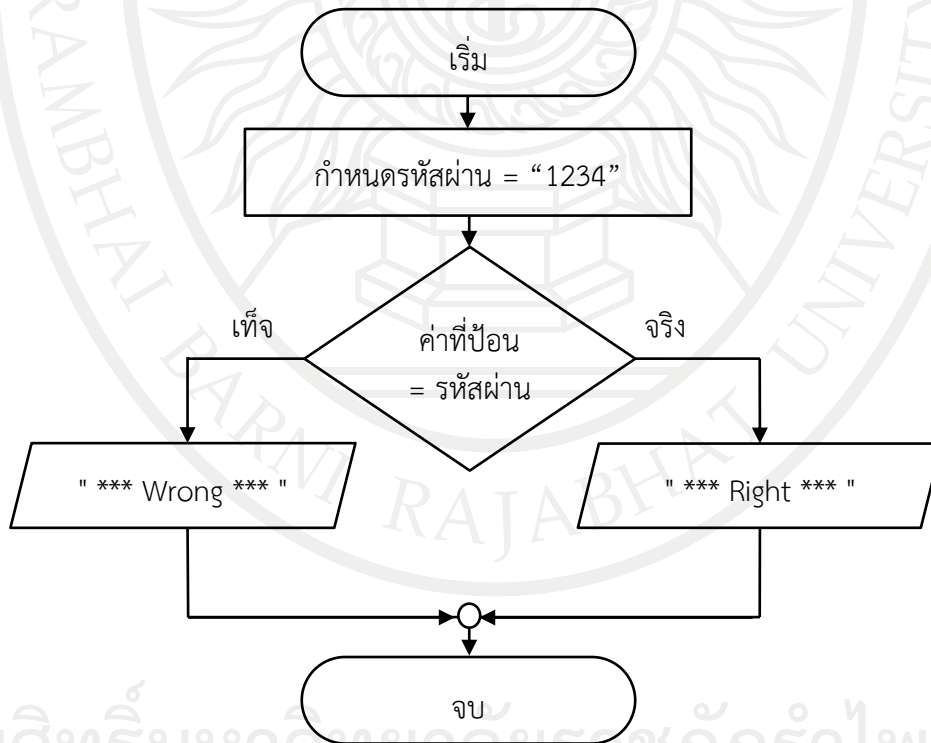
ภาพที่ 6.10 ผลลัพธ์โปรแกรมปุ่มสำหรับกด

ตัวอย่าง 6.5 การสร้างการป้อนรหัสผ่าน

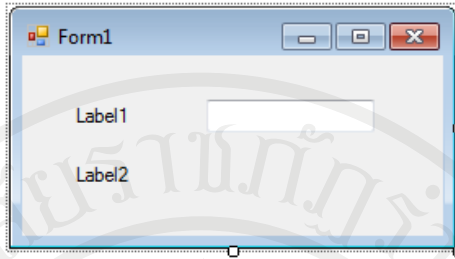
อัลกอริทึม

1. กำหนดรหัสผ่าน = "1234"
2. ตรวจสอบเงื่อนไขว่า ค่าที่ป้อนตรงกับรหัสผ่าน
 เงื่อนไขเป็นจริงให้ พิมพ์ข้อความว่า " *** Right *** "
 เงื่อนไขเป็นเท็จให้ พิมพ์ข้อความว่า " *** Wrong *** "

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.11



ภาพที่ 6.11 หน้าจอโปรแกรมการสร้างการป้อนรหัสผ่าน

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Dim password As Integer
```

```
    Dim answer As String = "1234"
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
        Label1.Text = "Password is "
```

```
        Label2.Text = " "
```

```
    End Sub
```

```
    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles TextBox1.TextChanged
```

```
        password = TextBox1.Text
```

```
        If password = answer Then
```

```
            Label2.Text = "*** Right ***"
```

```
        Else
```

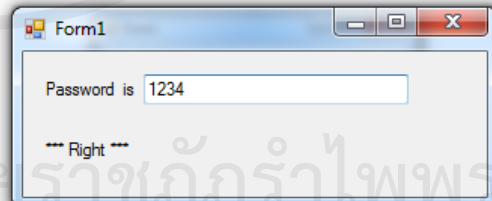
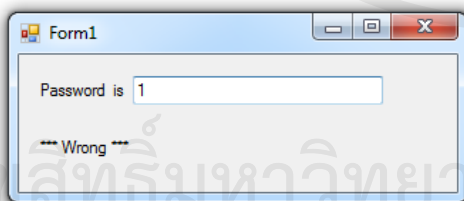
```
            Label2.Text = "*** Wrong ***"
```

```
        End If
```

```
    End Sub
```

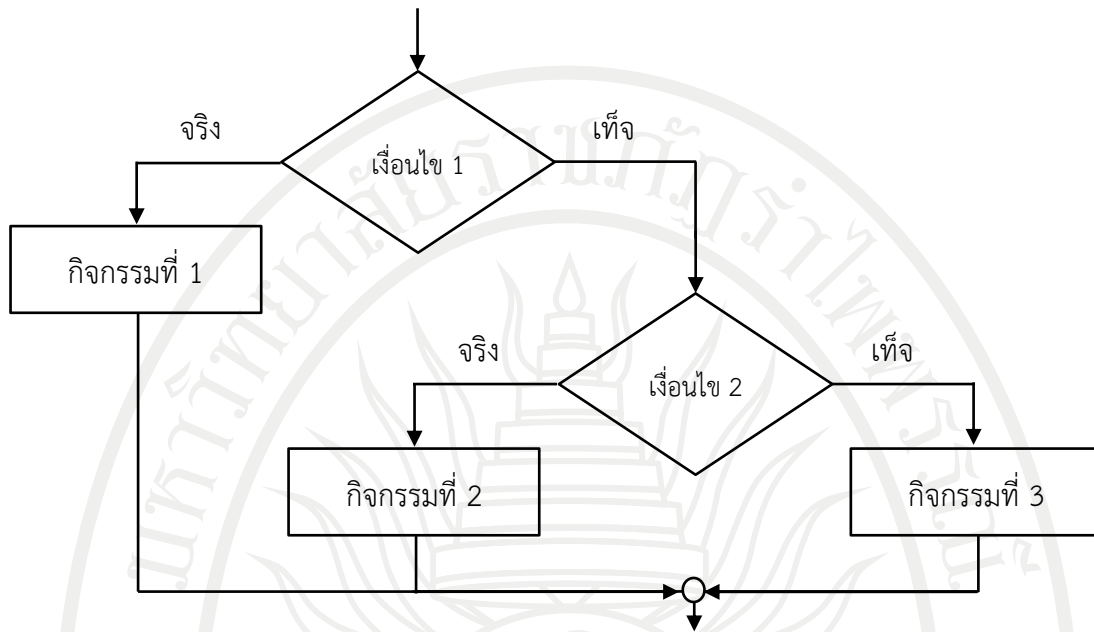
```
End Class
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.12



ภาพที่ 6.12 ผลลัพธ์โปรแกรมการสร้างการป้อนรหัสผ่าน

1.3 คำสั่ง If ... Then แบบที่ 3 แบบเงื่อนไขซ้อนเงื่อนไข



ภาพที่ 6.13 ผังงานแบบเลือกเส้นทางด้วยคำสั่ง If ... Then แบบที่ 3

จากภาพที่ 6.13 สามารถ เขียนคำสั่ง If ... Then ได้ดังนี้

```

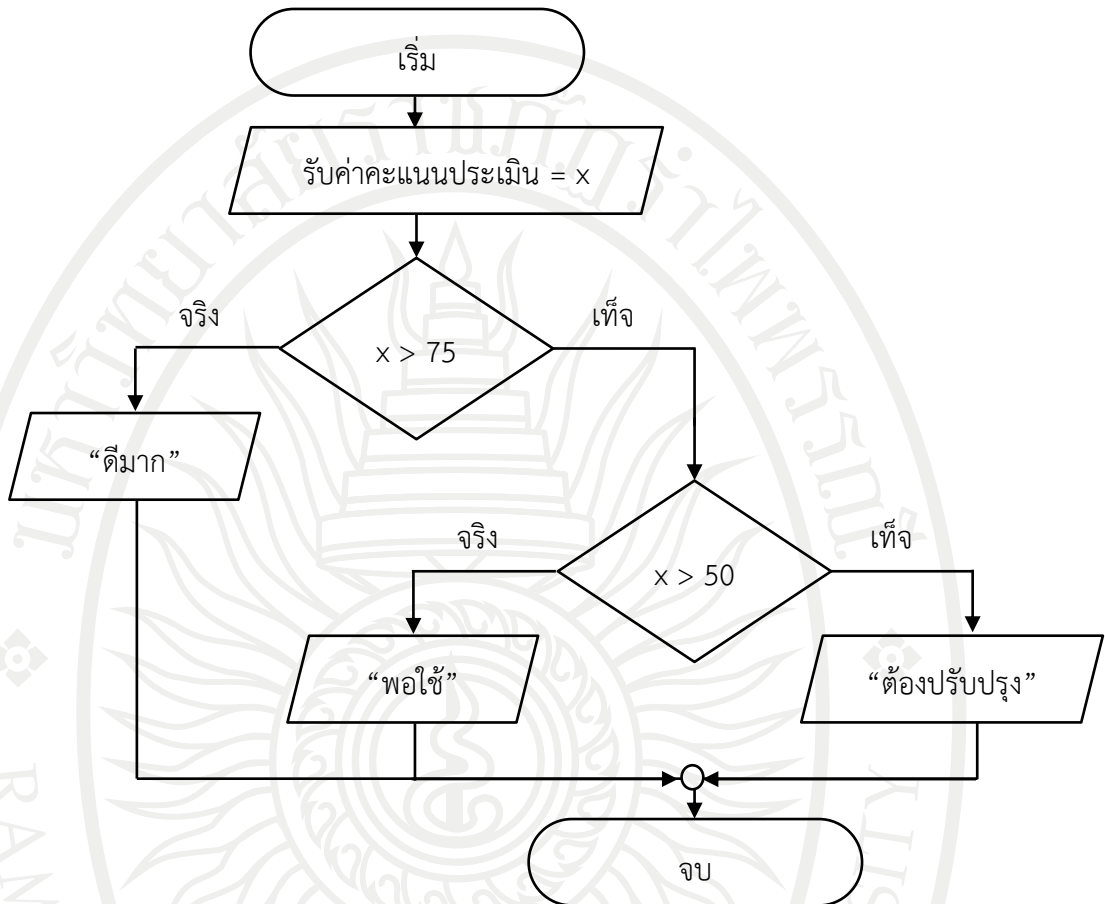
If เงื่อนไข 1 Then
    กิจกรรมที่ 1
Elseif เงื่อนไข 2 Then
    กิจกรรมที่ 2
Else
    กิจกรรมที่ 3
End If
  
```

ตัวอย่าง 6.6 การประเมินผลงาน

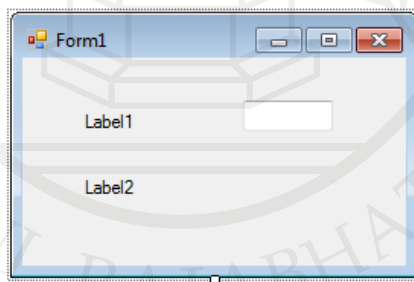
อัลกอริทึม

1. รับค่าคะแนนประเมิน
2. ตรวจสอบเงื่อนไขว่า ค่าคะแนนประเมิน > 75
 - เงื่อนไขเป็นจริงให้พิมพ์ข้อความว่า “ดีมาก”
 - เงื่อนไขเป็นเท็จให้ ตรวจสอบเงื่อนไขว่า ค่าคะแนนประเมิน > 50
 - เงื่อนไขเป็นจริงให้พิมพ์ข้อความว่า “พอใช้”
 - เงื่อนไขเป็นเท็จให้พิมพ์ข้อความว่า “ต้องปรับปรุง”

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.14



ภาพที่ 6.14 หน้าจอโปรแกรมการประเมินผลงาน

โปรแกรมคำสั่ง

Public Class Form1

Dim x As Integer

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _

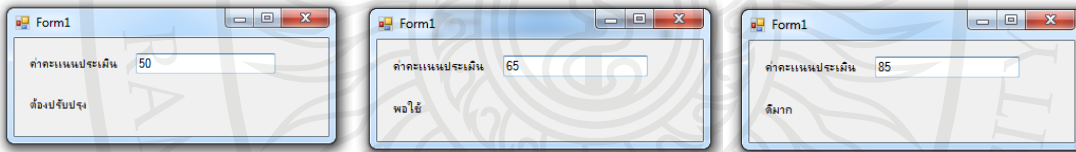
System.EventArgs) Handles MyBase.Load

```

Label1.Text = "ค่าคะแนนประเมิน"
Label2.Text = " "
End Sub
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles TextBox1.TextChanged
    x = TextBox1.Text
    If x > 75 Then
        Label2.Text = "ดีมาก"
    ElseIf x > 50 Then
        Label2.Text = "พอใช้"
    Else
        Label2.Text = "ต้องปรับปรุง"
    End If
End Sub
End Sub
End Class

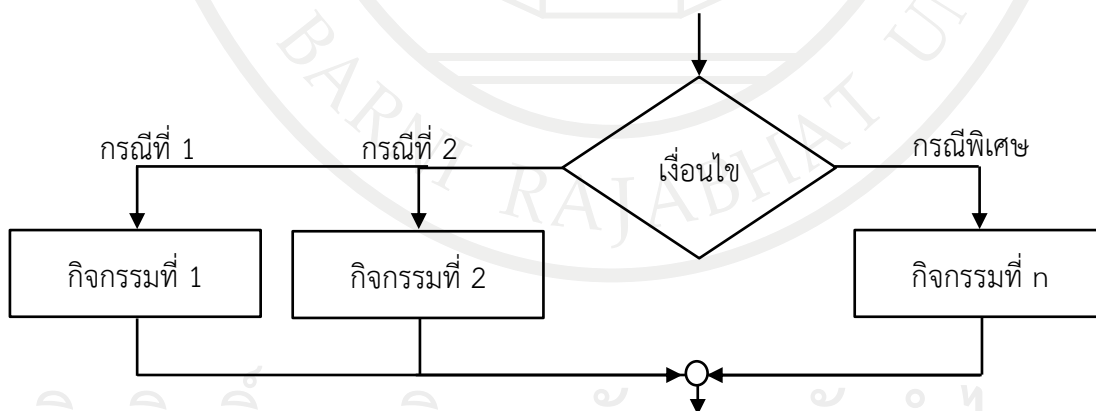
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.15



ภาพที่ 6.15 ผลลัพธ์โปรแกรมการประเมินผลงาน

2. คำสั่ง Select ... Case เป็นคำสั่งในการเลือกทำคำสั่งใดคำสั่งหนึ่งเท่านั้น โดยตรวจสอบค่าข้อมูลที่นำมาเปรียบเทียบตรงกับเงื่อนไขใด ถ้าเป็นจริงก็จะเข้ากรณีนั้น



ภาพที่ 6.16 ผังงานแบบเลือกเส้นทางด้วยคำสั่ง Select ... Case

จากภาพที่ ภาพที่ 6.16 สามารถ เขียนคำสั่ง Select ... Case ได้ดังนี้

```

Select Case ตัวแปรเปรียบเทียบ
Case กรณีที่ 1
    กิจกรรมที่ 1
Case กรณีที่ 2
    กิจกรรมที่ 2
Case Else
    กิจกรรมที่พิเศษ n
End Select
  
```

หมายเหตุ Case Else หมายถึงตัวแปรไม่ตรงกับกรณีใดเลย

กิจกรรมพิเศษหมายถึงกิจกรรมที่ทำเมื่อตัวแปรไม่ตรงกับกรณีใดเลย

ตัวอย่าง 6.7 รับค่าตัวเลขและแสดงข้อความ

อัลกอริทึม

1. รับค่าตัวเลข

2. ตรวจสอบเงื่อนไข

ถ้าใช่เลข 1

ให้แสดงข้อความว่า “หมายเลข หนึ่ง”

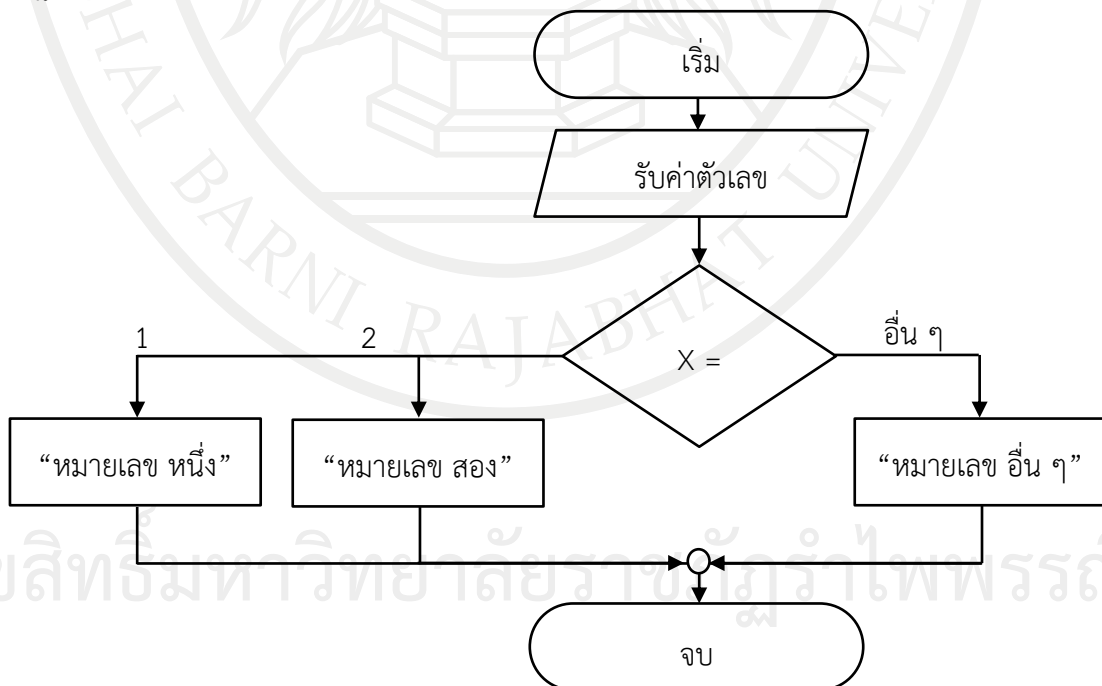
ถ้าใช่เลข 2

ให้แสดงข้อความว่า “หมายเลข สอง”

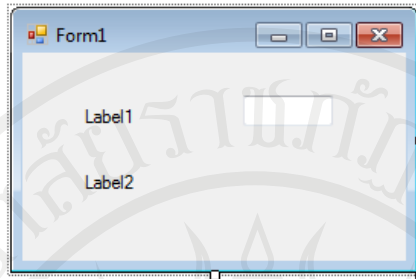
ถ้าไม่ใช่เลขตามเงื่อนไข

ให้แสดงข้อความว่า “หมายเลข อื่น ๆ”

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.17



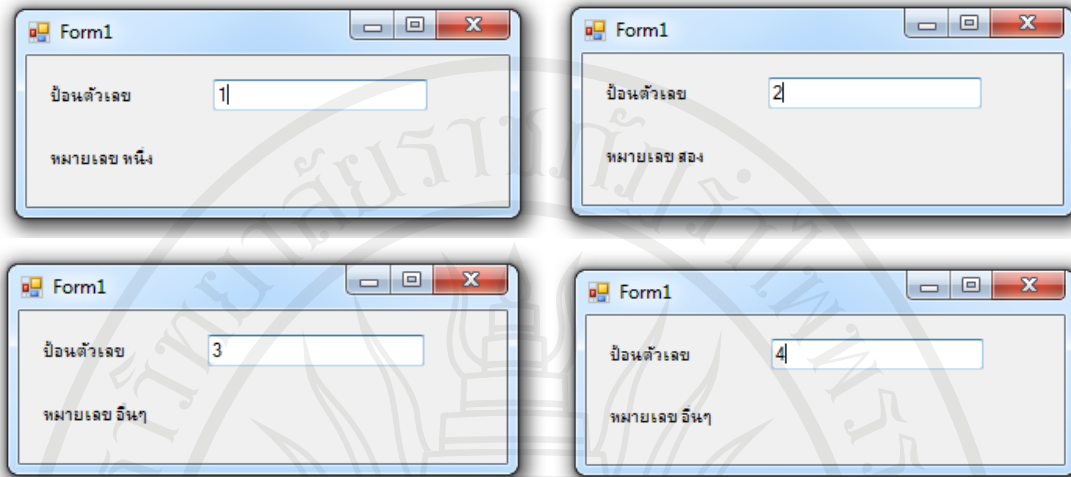
ภาพที่ 6.17 หน้าจอการรับค่าตัวเลขและแสดงข้อความ

โปรแกรมคำสั่ง

```
Public Class Form1
    Dim x As Integer
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load
        Label1.Text = "ป้อนตัวเลข"
        Label2.Text = " "
    End Sub
    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles TextBox1.TextChanged
        x = TextBox1.Text
        Select x
            Case 1
                Label2.Text = "หมายเลข หนึ่ง"
            Case 2
                Label2.Text = "หมายเลข สอง"
            Case Else
                Label2.Text = "หมายเลข อื่น ๆ "
        End Select
    End Sub
End Class
```

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 6.18

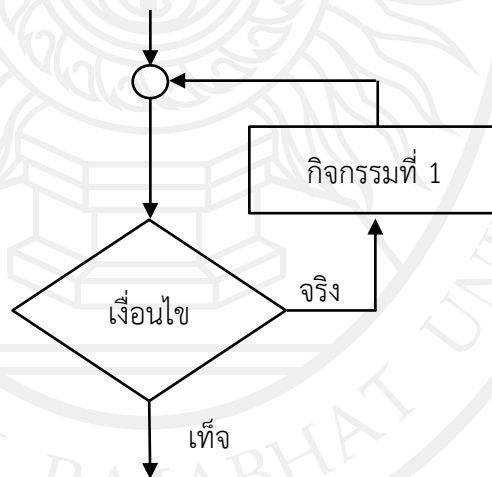


ภาพที่ 6.18 ผลลัพธ์โปรแกรมการรับค่าตัวเลขและแสดงข้อความ

คำสั่งควบคุมการทำงานแบบทำซ้ำ

เป็นโครงสร้างที่มีการทำงานแบบทำซ้ำคำสั่ง คือให้กลับมาทำงานในขั้นตอนอย่างเก่า จนกว่าจะตรงตามเงื่อนไขที่กำหนด ซึ่งลักษณะการวนประกอบไปด้วยกลุ่มคำสั่งที่เกี่ยวข้องดังนี้

1. คำสั่ง For ... Next



ภาพที่ 6.19 ผังงานแบบวนซ้ำด้วยคำสั่ง For ... Next

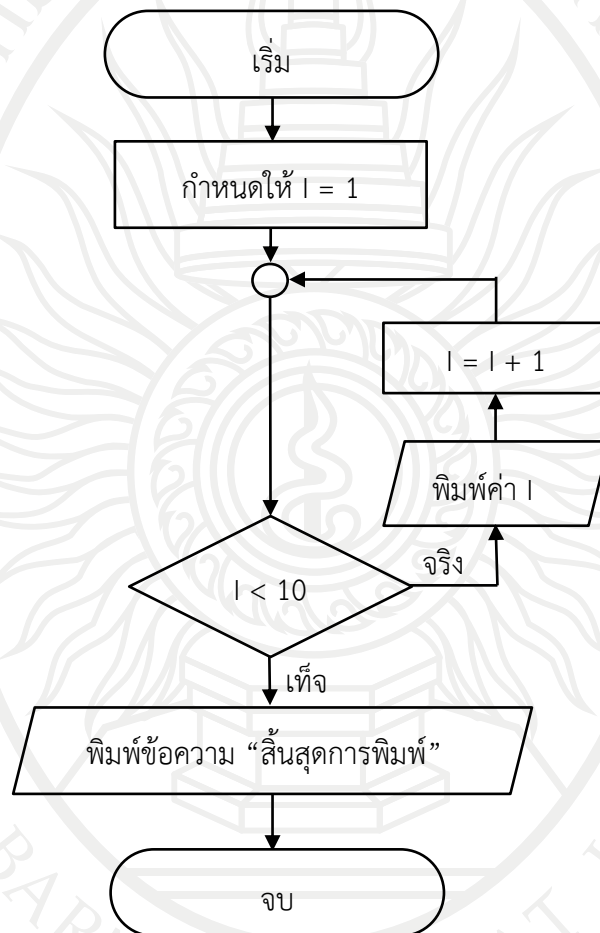
จากภาพที่ 6.19 สามารถเขียนคำสั่ง For ... Next ได้ดังนี้

For ตัวแปรนับ = ค่าที่เริ่มนับ To ค่าสิ้นสุดของตัวนับ Step จำนวนนับที่เพิ่มขึ้น
 กิจกรรมที่ 1
 Next ตัวแปรนับ

ตัวอย่างที่ 6.8 การพิมพ์เลข 1-10 ลงบนคอนโทรลทาบ็อกซ์
อัลกอริทึม

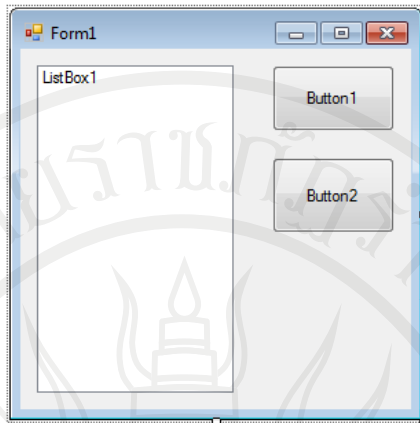
1. กำหนดให้ $i = 1$
2. ทำซ้ำจนกระทั่ง i มากกว่า 10 โดยเพิ่มค่าที่ละ 1
พิมพ์ค่า i
3. แสดงข้อความ “สิ้นสุดการพิมพ์”

ผังงาน



ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ออกแบบหน้าจอตั้งภาพที่ 6.20



ภาพที่ 6.20 หน้าจอโปรแกรมการพิมพ์เลข 1-10 ลงบนคอนโทรลลิสต์บ็อกซ์

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
        Dim i As Integer
```

```
        For i = 1 To 10 Step 1
```

```
            ListBox1.Items.Add(i)
```

```
        Next i
```

```
        ListBox1.Items.Add("จบการทำงาน")
```

```
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

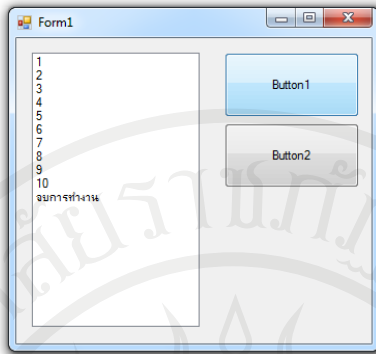
```
        ListBox1.Items.Clear()
```

```
    End Sub
```

```
End Class
```

เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.21

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



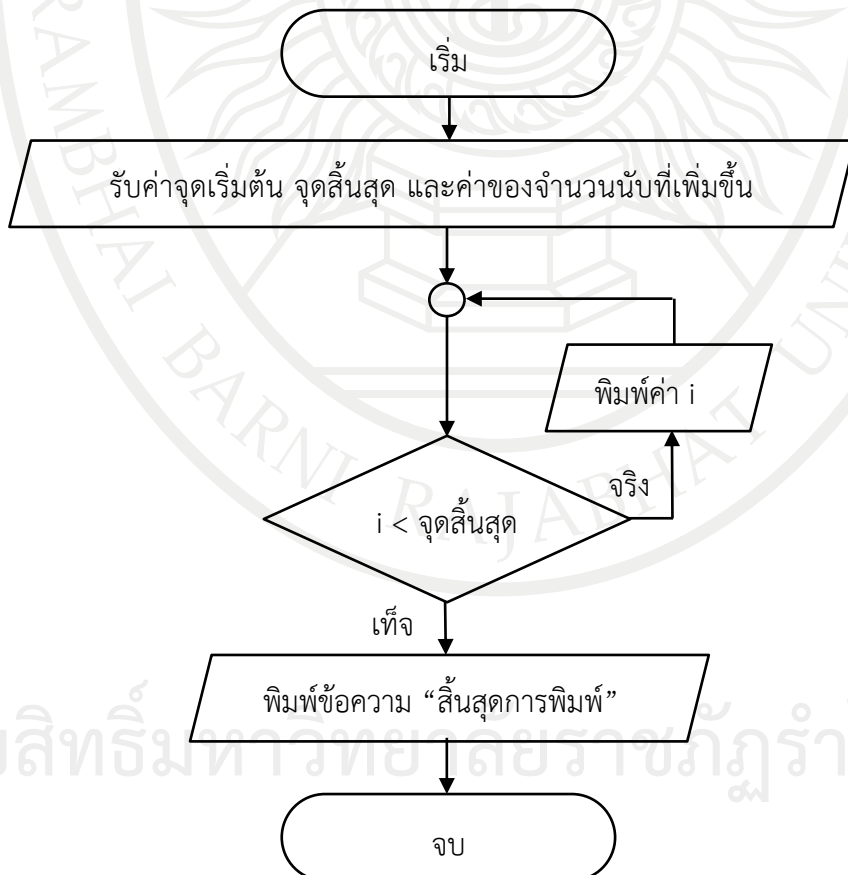
ภาพที่ 6.21 ผลลัพธ์หน้าจอโปรแกรมการพิมพ์เลข 1-10 ลงบนคอนโทรลลิตบ็อกซ์

ตัวอย่างที่ 6.9 การพิมพ์เลขตามข้อมูล จุดเริ่มต้น จุดสิ้นสุดและค่าของจำนวนนับที่เพิ่มขึ้นผ่านทางคอนโทรลเท็กซ์บ็อกซ์ลงบนคอนโทรลลิตบ็อกซ์

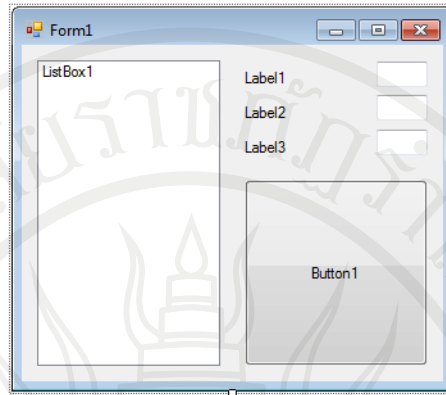
อัลกอริทึม

1. ให้รับค่าจุดเริ่มต้น จุดสิ้นสุด และค่าของจำนวนนับที่เพิ่มขึ้นผ่านทางคอนโทรลเท็กซ์บ็อกซ์
2. ทำซ้ำจนกระทั่ง i มากกว่า จุดสิ้นสุด
พิมพ์ค่า i
3. แสดงข้อความ “สิ้นสุดการพิมพ์”

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.22



ภาพที่ 6.22 หน้าจอการพิมพ์เลขตามข้อมูล จุดเริ่มต้น จุดสิ้นสุดและค่าของจำนวนนับที่เพิ่มขึ้นผ่านทางคอนโทรลเท็กซ์บ็อกซ์ลงบนคอนโทรลลิสต์บ็อกซ์

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        Label1.Text = "ค่าเริ่มต้น"
```

```
        Label2.Text = "ค่าสิ้นสุด"
```

```
        Label3.Text = "จำนวนนับที่เพิ่มขึ้น"
```

```
        Button1.Text = "แสดงผล"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
        Dim i As Integer
```

```
        ListBox1.Items.Clear()
```

```
        For i = TextBox1.Text To TextBox2.Text Step TextBox3.Text
```

```
            ListBox1.Items.Add(i)
```

```
        Next i
```

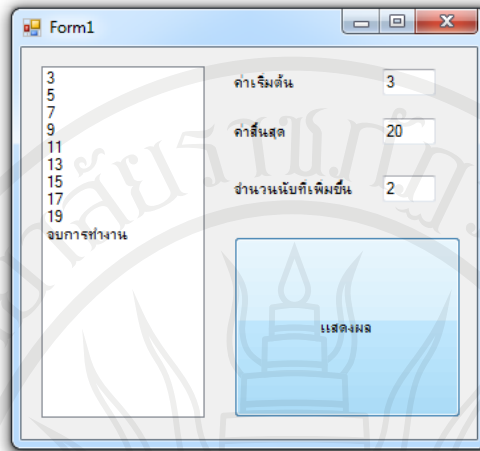
```
        ListBox1.Items.Add("จบการทำงาน")
```

```
    End Sub
```

```
End Class
```

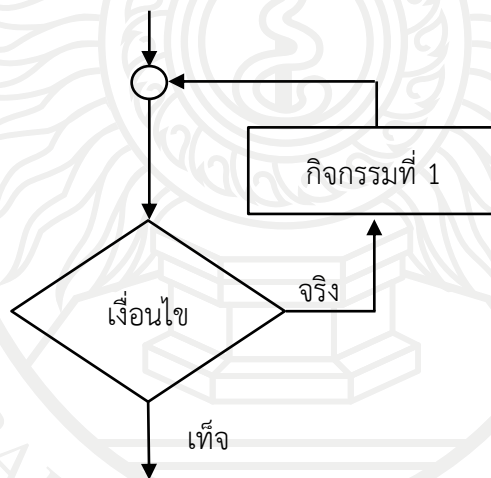
ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.23



ภาพที่ 6.23 ผลลัพธ์การพิมพ์เลขตามข้อมูล จุดเริ่มต้น จุดสิ้นสุดและค่าของจำนวนนับที่เพิ่มขึ้นผ่านทางคอนโทรลที่กซ์บ็อกซ์ลงบนคอนโทรลลิสต์บ็อกซ์

2. คำสั่ง Do While ... Loop



ภาพที่ 6.24 ผังงานแบบวนซ้ำด้วยคำสั่ง Do While ... Loop

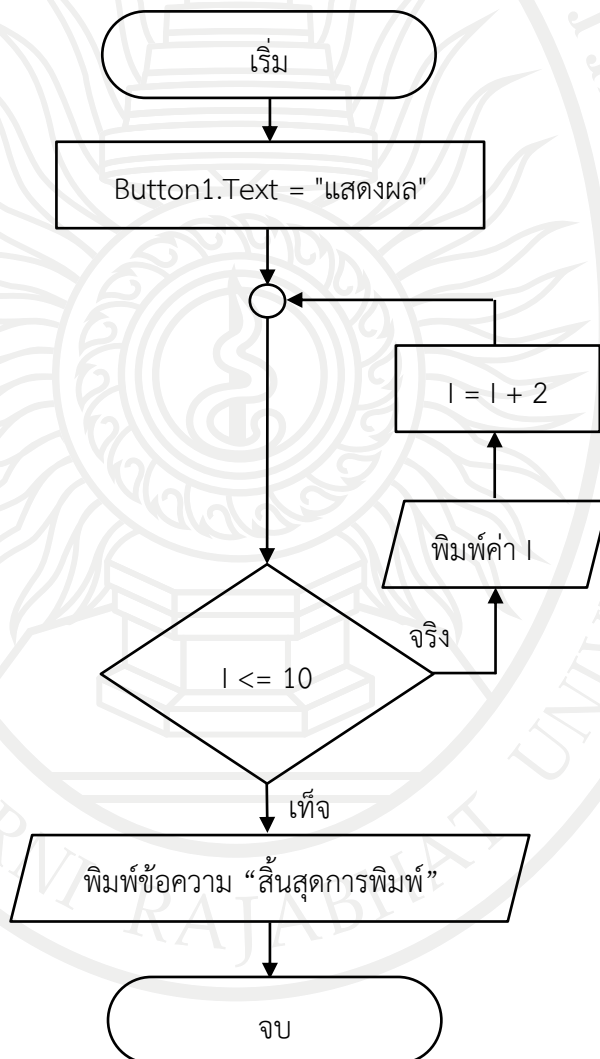
จากภาพที่ 6.24 สามารถ เขียน Do While ... Loop ได้ดังนี้

Do While เงื่อนไข
กิจกรรมที่ 1
Loop

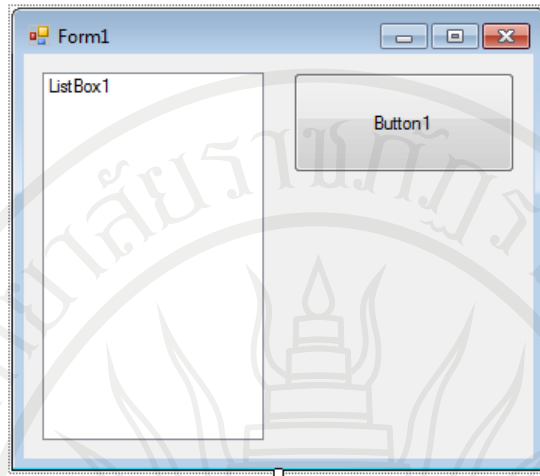
ตัวอย่างที่ 6.10 การพิมพ์เลข 1-10 ลงบนคอนโทรลลิทบ็อกซ์โดยนับเพิ่มขึ้นทีละ 2
อัลกอริทึม

1. กำหนดให้ปุ่มกดมีเพียงปุ่มเดียว และแสดงข้อความว่า "แสดงผล"
2. ทำซ้ำจนกระทั่ง i ไม่ได้น้อยกว่าหรือเท่ากับ 10
พิมพ์ค่า i
 $i = i + 2$
3. แสดงข้อความ "สิ้นสุดการพิมพ์"

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.25



ภาพที่ 6.25 หน้าจอการพิมพ์เลข 1-10 ลงบนคอนโทรลลิสต์บ็อกซ์โดยนับเพิ่มขึ้นทีละ 2

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        Button1.Text = "แสดงผล"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
        Dim i As Integer = 1
```

```
        ListBox1.Items.Clear()
```

```
        Do While i <= 10
```

```
            ListBox1.Items.Add(i)
```

```
            i = i + 2
```

```
        Loop
```

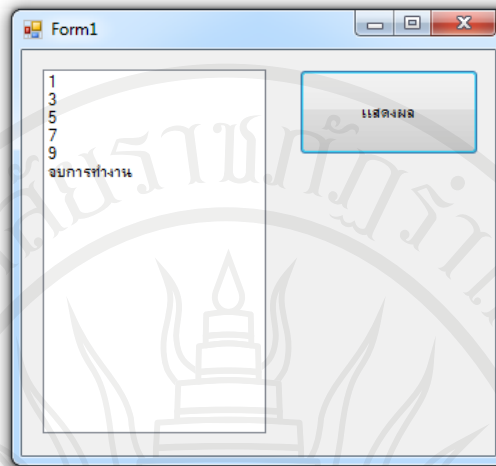
```
        ListBox1.Items.Add("จบการทำงาน")
```

```
    End Sub
```

```
End Class
```

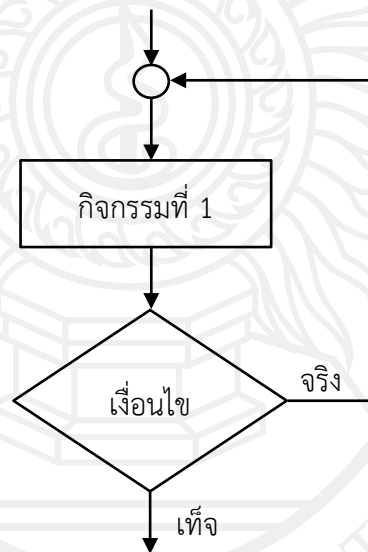
ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.26



ภาพที่ 6.26 ผลลัพธ์การพิมพ์เลข 1-10 ลงบนคอนโทรลลิสต์โดยนับเพิ่มขึ้นทีละ 2

3. คำสั่ง Do ... Loop While



ภาพที่ 6.27 ผังงานแบบวนซ้ำด้วยคำสั่ง Do ... Loop While

จากภาพที่ 6.27 สามารถ เขียนคำสั่ง Do ... Loop While ได้ดังนี้

Do

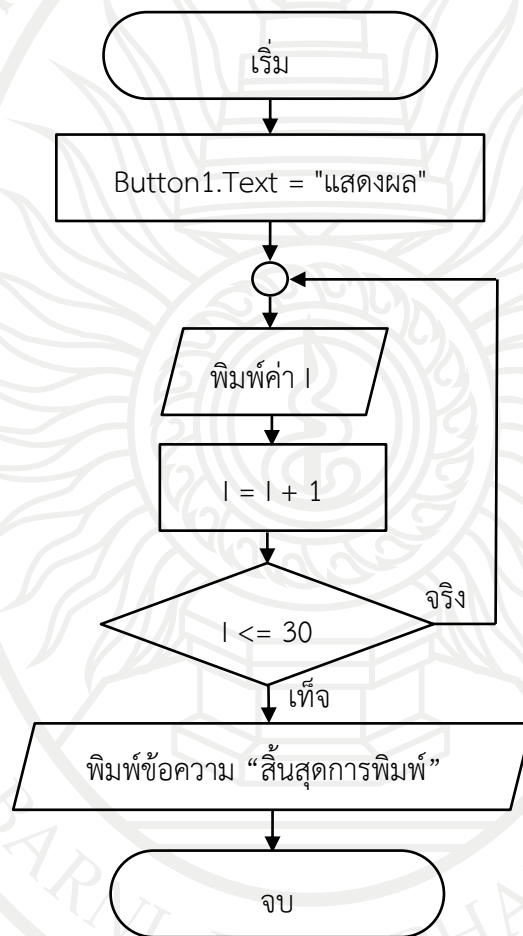
 กิจกรรมที่ 1

Loop While เงื่อนไข

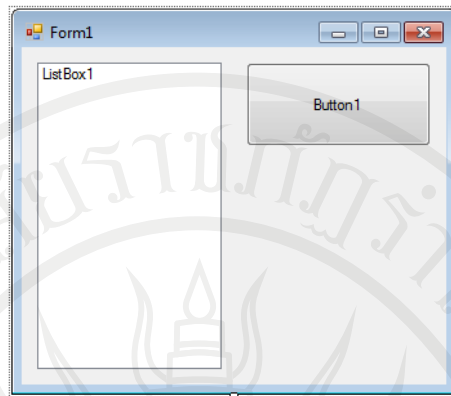
ตัวอย่างที่ 6.11 การพิมพ์เลข 1-30 ลงบนคอนโทรลทึบ็อกซ์
อัลกอริทึม

1. กำหนดให้ปุ่มกดมีเพียงปุ่มเดียว และแสดงข้อความว่า "แสดงผล"
2. ทำซ้ำจนกระทั่ง i ไม่ได้น้อยกว่าหรือเท่ากับ 30
พิมพ์ค่า i
 $i = i + 1$
3. แสดงข้อความ "สิ้นสุดการพิมพ์"

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.28



ภาพที่ 6.28 หน้าจอโปรแกรมการพิมพ์เลข 1-30 ลงบนคอนโทรลลิสต์บ็อกซ์

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        Button1.Text = "แสดงผล"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
        Dim i As Integer = 1
```

```
        ListBox1.Items.Clear()
```

```
        Do
```

```
            ListBox1.Items.Add(i)
```

```
            i = i + 1
```

```
        Loop While i <= 30
```

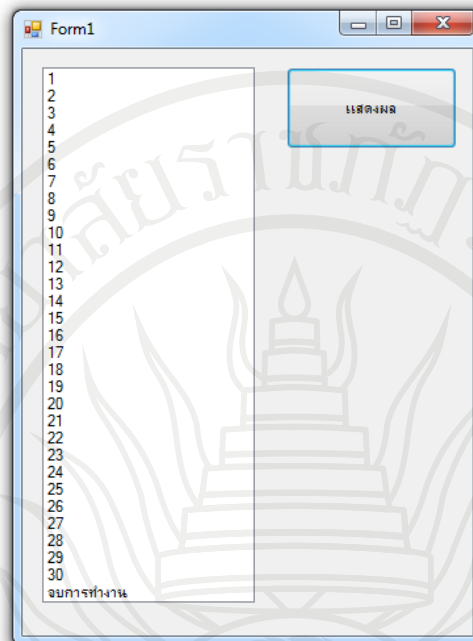
```
        ListBox1.Items.Add("จบการทำงาน")
```

```
    End Sub
```

```
End Class
```

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.29



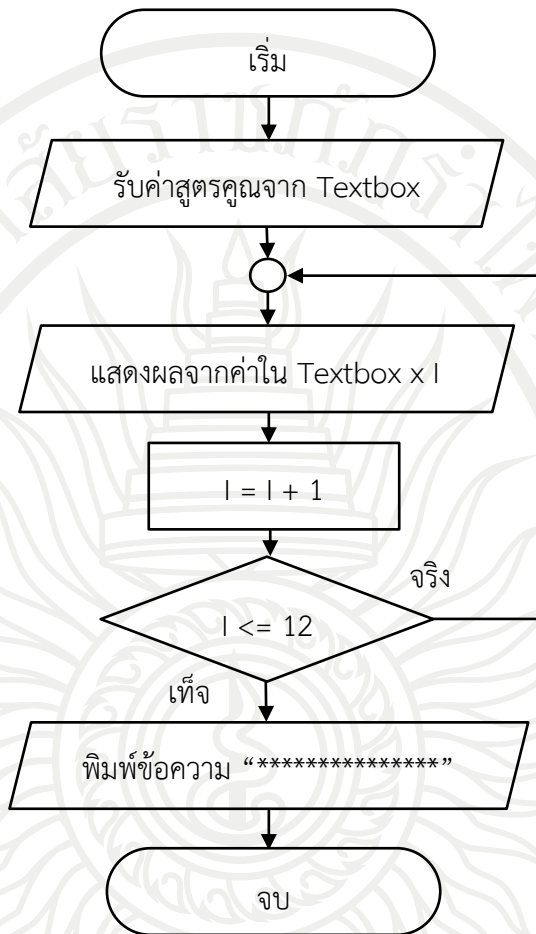
ภาพที่ 6.29 ผลลัพธ์โปรแกรมการพิมพ์เลข 1-30 ลงบนคอนโทรลลิทบ็อกซ์

ตัวอย่าง 6.12 การพิมพ์สูตรคูณลงบนคอนโทรลลิทบ็อกซ์
อัลกอริทึม

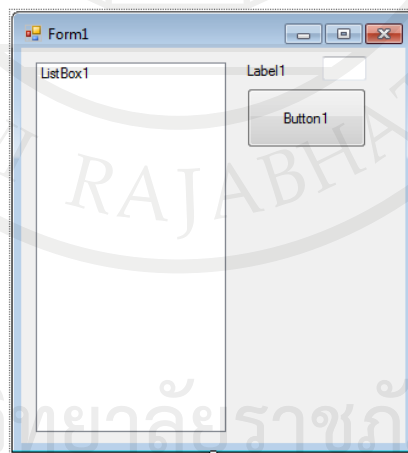
- 1 รับค่าสูตรคูณจาก Textbox
- 2 แสดงผล เมื่อนำค่าใน Textbox x l
- 3 ให้ $l = l + 1$
- 4 ทำซ้ำจนกว่า l ไม่ได้น้อยกว่าหรือเท่ากับ 12
- 5 แสดงข้อความ "*****"

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ผังงาน



ออกแบบหน้าจอ ดังภาพที่ 6.30



ภาพที่ 6.30 หน้าจอโปรแกรมการพิมพ์สูตรคูณลงบนคอนโทรลลิสต์บ็อกซ์

โปรแกรมคำสั่ง

Public Class Form1

Dim num As Integer

Dim sum As Integer

Dim i As Integer = 1

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label1.Text = "สูตรคูณแม่ : "

Button1.Text = "แสดงผล"

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

num = TextBox1.Text

Do

sum = num * i

ListBox1.Items.Add(i & " x " & num & " = " & sum)

i = i + 1

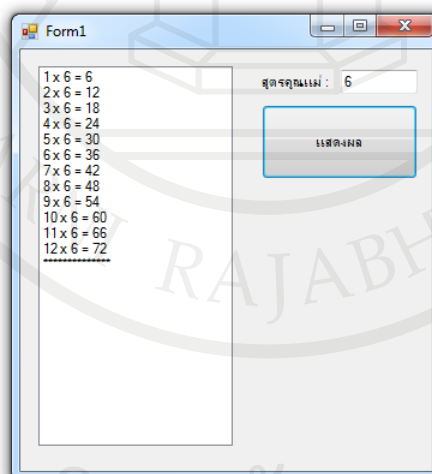
Loop While i <= 12

ListBox1.Items.Add("*****")

End Sub

End Class

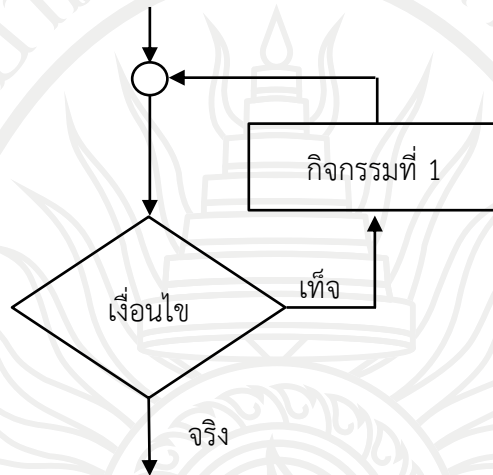
เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.31



ภาพที่ 6.31 ผลลัพธ์โปรแกรมการพิมพ์สูตรคูณลงบนคอนโทรลลิทบ็อกซ์

ข้อแตกต่างของลูป Do While ... Loop กับ Do ... Loop While อยู่ที่กรณีที่เป็นเท็จ ถ้าเป็น Do While ... Loop จะไม่ปรากฏข้อความใดเลย ซึ่งเรียกโครงสร้างแบบนี้ว่า Pre loop test ส่วน Do ... Loop While จะปรากฏค่าตัวเลขมาก่อน 1 ตัวแล้วจึงตรวจสอบเงื่อนไข เราเรียกโครงสร้างแบบนี้ว่า Post loop test

4. คำสั่ง Do Until ... Loop



ภาพที่ 6.32 ผังงานแบบวนซ้ำด้วยคำสั่ง Do Until ... Loop

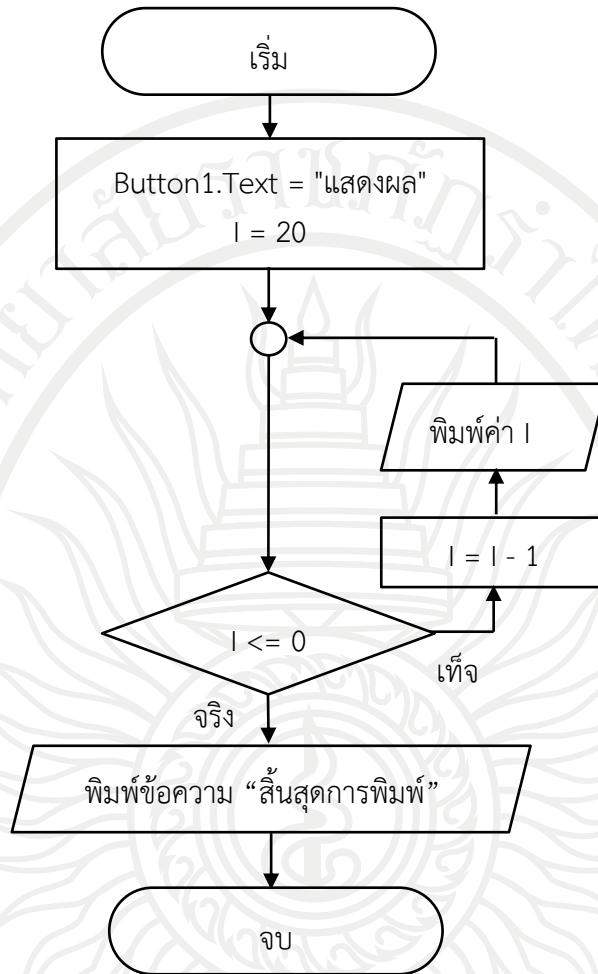
จากภาพที่ 6.32 สามารถ เขียน Do Until ... Loop ได้ดังนี้

Do Until เงื่อนไข
กิจกรรมที่ 1
Loop

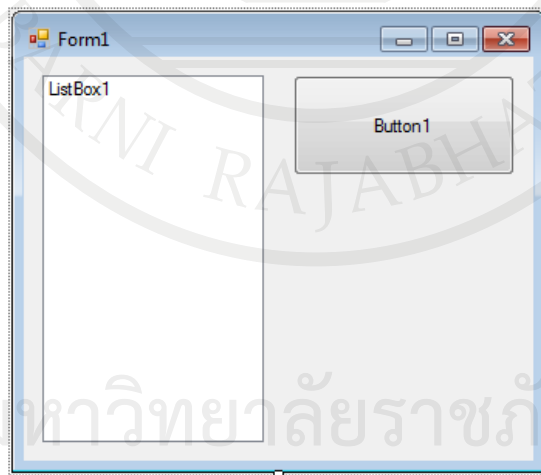
ตัวอย่างที่ 6.13 การพิมพ์เลข 20-1 ลงบนคอนโทรลลิทบ็อกซ์
อัลกอริทึม

1. กำหนดให้ปุ่มกดมีเพียงปุ่มเดียว และแสดงข้อความว่า "แสดงผล"
2. กำหนดให้ $i = 20$
3. ทำซ้ำจนกระทั่ง i น้อยกว่าหรือเท่ากับ 0
พิมพ์ค่า i
ให้ $i = i - 1$
4. แสดงข้อความ "สิ้นสุดการพิมพ์"

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.33



ภาพที่ 6.33 หน้าจอโปรแกรมการพิมพ์เลข 20-1 ลงบนคอนโทรลลิสต์บ็อกซ์

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles MyBase.Load
```

```
        Button1.Text = "แสดงผล"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles Button1.Click
```

```
        Dim i As Integer = 20
```

```
        ListBox1.Items.Clear()
```

```
        Do Until i <= 0
```

```
            ListBox1.Items.Add(i)
```

```
            i = i - 1
```

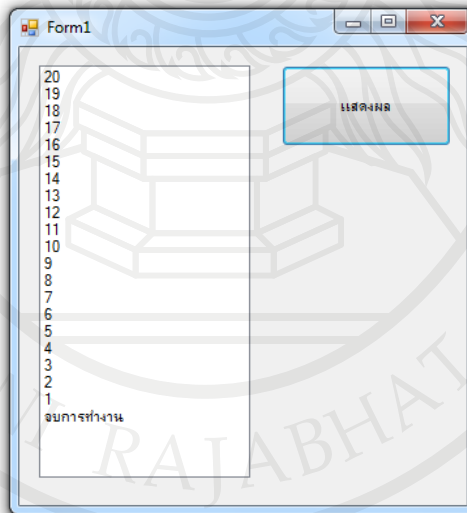
```
        Loop
```

```
        ListBox1.Items.Add("จบการทำงาน")
```

```
    End Sub
```

```
End Class
```

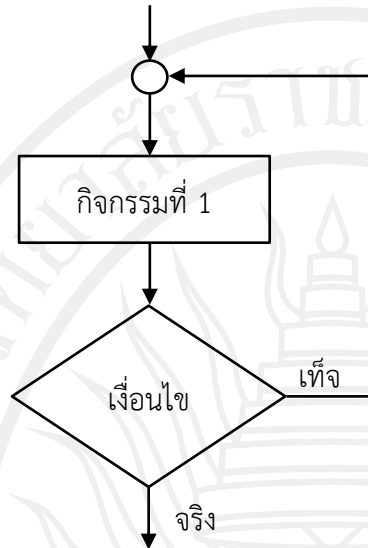
เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.31



ภาพที่ 6.34 ผลลัพธ์โปรแกรมการพิมพ์เลข 20-1 ลงบนคอนโทรลลิสต์บ็อกซ์

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

5. คำสั่ง Do ... Loop Until



ภาพที่ 6.35 ผังงานแบบวนซ้ำด้วยคำสั่ง Do ... Loop Until

จากภาพที่ 6.35 สามารถ เขียน Do ... Loop Until ได้ดังนี้

```

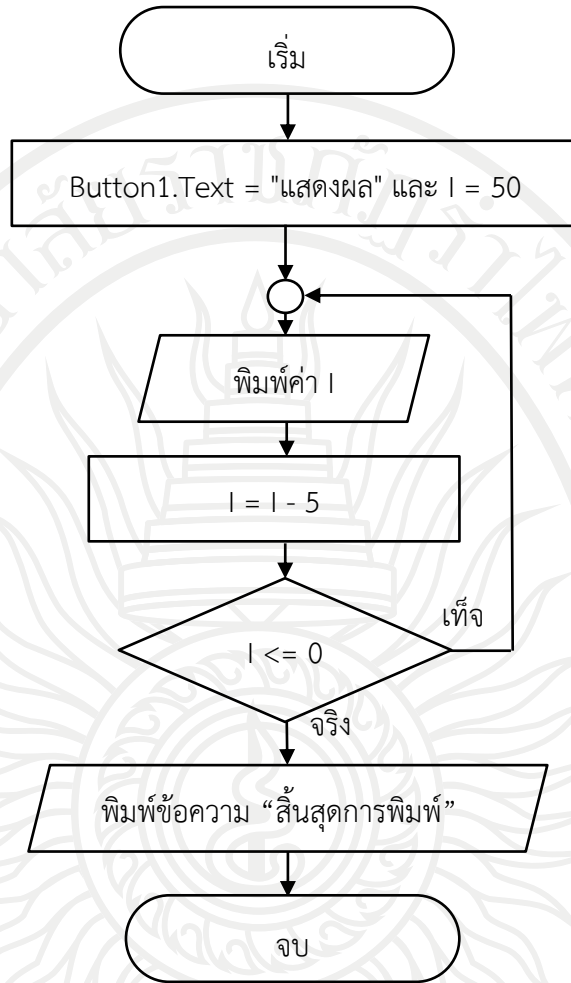
Do
    กิจกรรมที่ 1
Loop Until เงื่อนไข
  
```

ตัวอย่าง 6.14 การพิมพ์เลข 50-1 ลงบนคอนโทรลทาบ็อกซ์ โดยลดลงทีละ 5 อัลกอริทึม

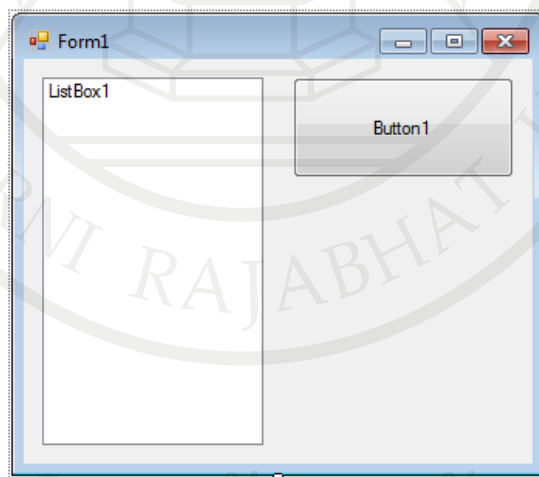
1. กำหนดให้ปุ่มกดมีเพียงปุ่มเดียว และแสดงข้อความว่า "แสดงผล"
2. กำหนดให้ $i = 50$
3. ทำซ้ำจนกระทั่ง i น้อยกว่าหรือเท่ากับ 0
 - พิมพ์ค่า i
 - ให้ $i = i - 5$
4. แสดงข้อความ "สิ้นสุดการพิมพ์"

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

ผังงาน



ออกแบบหน้าจอตั้งภาพที่ 6.36



ภาพที่ 6.36 หน้าจอโปรแกรมการพิมพ์เลข 50-1 ลงบนคอนโทรลลิสต์บ็อกซ์ โดยลดลงทีละ 5

โปรแกรมคำสั่ง

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles MyBase.Load
```

```
        Button1.Text = "แสดงผล"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles Button1.Click
```

```
        Dim i As Integer = 50
```

```
        ListBox1.Items.Clear()
```

```
        Do
```

```
            ListBox1.Items.Add(i)
```

```
            i = i - 5
```

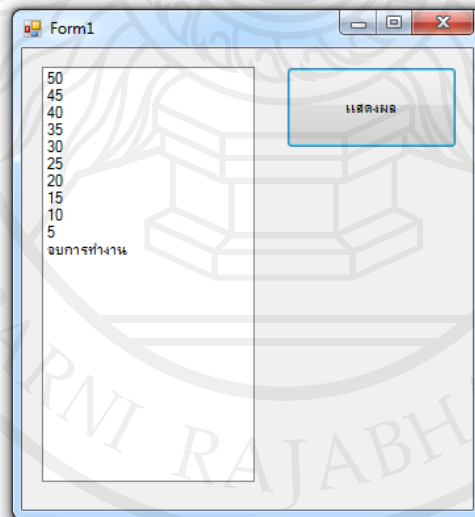
```
        Loop Until i <= 0
```

```
        ListBox1.Items.Add("จบการทำงาน")
```

```
    End Sub
```

```
End Class
```

เมื่อกด F5 จะได้ผลลัพธ์ดังภาพที่ 6.37



ภาพที่ 6.37 ผลลัพธ์โปรแกรมการพิมพ์เลข 50-1 ลงบนคอนโทรลลิสต์บ็อกซ์ โดยลดลงทีละ 5

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

สรุป

สภาพปัญหาที่อาจเกิดขึ้นมีหลายรูปแบบ การเขียนโปรแกรมก็มีหลายรูปแบบซึ่งขึ้นอยู่กับความซับซ้อนในสภาพปัญหานั้น ๆ โดยการเขียนคำสั่งควบคุมการทำงานแบบเป็นประเภทใหญ่ ๆ ได้ 3 อย่างคือ 1) คำสั่งควบคุมการทำงานแบบลำดับเป็นการควบคุมการทำงานทีละคำสั่งจากบนลงล่าง รวมถึงการรับข้อมูลเข้าและการเอาข้อมูลแสดงผลออก 2) คำสั่งควบคุมการทำงานแบบเลือกเส้นทางเป็นการทำงานที่มีการตัดสินใจโดยมีเงื่อนไขเป็นตัวกำหนด เป็นกลุ่มคำสั่งที่ต้องใช้มากที่สุดในการเขียนโปรแกรม ไม่ว่าจะเป็นภาษาใดก็ตาม โดยกลุ่มคำสั่งที่เกี่ยวกับการสร้างเงื่อนไขแบ่งได้เป็น 2 คำสั่งใหญ่ ๆ คือ If ... Then และ คำสั่ง Select ... Case 3) คำสั่งควบคุมการทำงานแบบทำซ้ำเป็นการทำงานแบบทำซ้ำคำสั่งคือให้กลับมาทำงานในขั้นตอนอย่างเก่า จนกว่าจะตรงตามเงื่อนไขที่กำหนด ซึ่งลักษณะการวนประกอบไปด้วยกลุ่มคำสั่งได้แก่ คำสั่ง For ... Next , คำสั่ง Do While ... Loop, คำสั่ง Do ... Loop While , คำสั่ง Do Until ... Loop และคำสั่ง Do ... Loop Until ทั้งนี้ นักศึกษาสามารถหาข้อมูลตัวอย่างวิธีการเขียนโปรแกรมคำสั่งและการใช้คำสั่งเพิ่มเติมได้ที่หนังสือ Microsoft Visual Basic 2010 Step by Step. (Halvorson, M. 2010). และ Microsoft Visual Basic 2010 Developer's Handbook, (Loffelmann, K., & CallaPurohit ,S.2010).

แบบฝึกหัดบทที่ 6

1. จงเขียนโปรแกรมเพื่อคำนวณหาพื้นที่สามเหลี่ยมด้านเท่า
2. จงยกตัวอย่าง โจทย์ If ... Then แบบเงื่อนไขซ้อนเงื่อนไขมา 2 โจทย์
3. จงยกตัวอย่าง โจทย์ For ... Next มา 2 โจทย์ พร้อมทั้งเขียนโปรแกรมคำสั่ง
4. ข้อแตกต่างของรูป Do While ... Loop กับ Do ... Loop While คืออะไร อธิบายพร้อมยกตัวอย่าง
5. ในกลุ่มของคำสั่งควบคุมการทำงานแบบเลือกเส้นทาง เช่น IF มีวิธีการเขียนกี่รูปแบบ
6. ในกลุ่มของคำสั่งควบคุมการทำงานแบบทำซ้ำมีวิธีการเขียนกี่รูปแบบ
7. จงเขียนโปรแกรมการตรวจสอบคะแนนที่ป้อนเข้ามา ถ้าได้คะแนน 60 ขึ้นไป จะมีผลลัพธ์คือ "ผ่าน" แต่หากต่ำกว่า 60 จะมีผลลัพธ์คือ "ไม่ผ่าน"
8. จงเขียนโปรแกรมตัดเกรด แบบ 5 เกรด (A-F) ตามเงื่อนไขของคะแนนคือ ถ้าคะแนนตั้งแต่ 80 ขึ้นไปผลลัพธ์คือ A , ถ้าคะแนนตั้งแต่ 70 ขึ้นไปผลลัพธ์คือ B ถ้าคะแนนตั้งแต่ 60 ขึ้นไปผลลัพธ์คือ C , ถ้าคะแนนตั้งแต่ 50 ขึ้นไปผลลัพธ์คือ D ถ้าน้อยกว่า 50 ผลลัพธ์คือ F , และหากไม่ตรงเงื่อนไขใด ๆ ผลลัพธ์คือ Error
9. จงอธิบายโปรแกรมคำสั่งต่อไปนี้

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim i As Integer
    ListBox1.Items.Clear()
    Do Until i >= 10
        ListBox1.Items.Add(i)
        i = i + 3
    Loop
    ListBox1.Items.Add("จบการทำงาน")
End Sub
```

10. จากโปรแกรมคำสั่งต่อไปนี้เมื่อกด F5 จะแสดงผลอะไรออกมา

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim i As Integer
    For i = 1 To 100 Step 5
        ListBox1.Items.Add(i)
    Next i
    ListBox1.Items.Add("จบการทำงาน")
End Sub
```


เอกสารอ้างอิง

- ธีรวัฒน์ ประกอบผล, (2554). คู่มือการเขียนแอปพลิเคชันด้วย Visual Basic 2010. กรุงเทพมหานคร: ซิมพลิฟาย.
- สัจจะ จรัสรุ่งรวีร, (2554). คู่มือเรียนและใช้งาน Visual Basic 2010. นนทบุรี: ไอทีซี พรีเมียร์จำกัด.
- Halvorson, M. (2010). Microsoft Visual Basic 2010 Step by Step. United States of America.
- Loffelmann, K.,& CallaPurohit ,S.(2010). Microsoft Visual Basic 2010 Developer's Handbook, Canada: O' Reilly Media.

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

แผนบริหารการสอนประจำบทที่ 7

หัวข้อเนื้อหาบทที่ 7

1. โปรแกรมเครื่องคิดเลข
2. โปรแกรมตั้งเวลาปลุก
3. โปรแกรมแสดงตัวอักษรวิ่ง
4. โปรแกรมบอร์ดควบคุมระดับน้ำ
5. โปรแกรมบอร์ดควบคุมไฟในถังผสม
6. โปรแกรมควบคุมไฟจราจร
7. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 7 แล้วนักศึกษาสามารถ

1. สามารถออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรมได้
 2. สามารถนำความรู้ไปประยุกต์ใช้ในการออกแบบโปรแกรมสำหรับงานวิศวกรรมต่าง ๆ ได้
- เช่น โปรแกรมเครื่องคิดเลข, โปรแกรมตั้งเวลาปลุก, โปรแกรมแสดงตัวอักษรวิ่ง, โปรแกรมบอร์ดควบคุมระดับน้ำ, โปรแกรมบอร์ดควบคุมไฟในถังผสมและโปรแกรมควบคุมไฟจราจร เป็นต้น

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายเนื้อหา
2. ให้นักศึกษาได้เรียนรู้ผ่านการบรรยายที่จอโปรเจคเตอร์
3. ให้ผู้เรียนทำแบบฝึกหัดบทที่ 7

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชา 6091102 การโปรแกรมคอมพิวเตอร์
2. เครื่องฉายโปรเจคเตอร์
3. โปรแกรมคอมพิวเตอร์
4. หนังสืออ่านประกอบค้นคว้าเพิ่มเติม
5. แบบฝึกหัดบทที่ 7

การวัดผลและประเมินผล

1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากกิจกรรมร่วม
3. สังเกตจากความสนใจ
4. ประเมินจากการทำแบบฝึกหัด

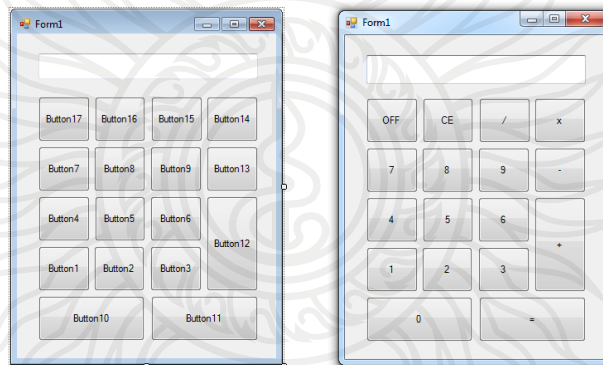
บทที่ 7

การออกแบบและพัฒนาโปรแกรมสำหรับงานวิศวกรรม

จากการรวบรวมเรียบเรียงเนื้อหาจาก คู่มือการเขียนแอปพลิเคชันด้วย Visual Basic 2010. (ธีรวัฒน์, 2554) คู่มือเรียนและใช้งาน Visual Basic 2010. (สัจจะ, 2554) ซึ่งเป็นวิธีการใช้งานโปรแกรม วิชาการเบสิกแล้วนั้น ทำให้ได้วิธีคิดและวิธีเขียนโปรแกรมด้วยคำสั่งต่าง ๆ เนื้อหาในบทนี้จึงเป็นตัวอย่าง ประยุกต์สำหรับใช้ในทางวิศวกรรม โดยใช้เป็นโจทย์โปรแกรมสำหรับศึกษาเรียนรู้ได้แก่โปรแกรมเครื่อง คณิตเลข, โปรแกรมตั้งเวลาปลุก, โปรแกรมแสดงอักษรวิ่ง, โปรแกรมบอร์ดควบคุมระดับน้ำ, โปรแกรม บอร์ดควบคุมไฟในถังผสมและโปรแกรมควบคุมไฟจราจร เพื่อเป็นตัวอย่างให้ผู้ที่สนใจได้นำไปใช้ ออกแบบและพัฒนาสำหรับงานวิศวกรรมต่อไป

โปรแกรมเครื่องคณิตเลข

ออกแบบโปรแกรมในการสร้างเครื่องคณิตเลขสำหรับคำนวณโดยใช้คำสั่งโอเปอเรเตอร์ทาง คณิตศาสตร์ +, -, x และ /



ภาพที่ 7.1 โปรแกรมเครื่องคณิตเลข

โปรแกรมคำสั่ง

Public Class Form1

Dim operators As String

Dim num1 As Integer

Dim num2 As Integer

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _

System.EventArgs) Handles MyBase.Load

Button1.Text = "1"

Button2.Text = "2"

Button3.Text = "3"

Button4.Text = "4"

Button5.Text = "5"

```

Button6.Text = "6"
Button7.Text = "7"
Button8.Text = "8"
Button9.Text = "9"
Button10.Text = "0"
Button11.Text = "="
Button12.Text = "+"
Button13.Text = "-"
Button14.Text = "x"
Button15.Text = "/"
Button16.Text = "CE"
Button17.Text = "OFF"
TextBox1.TextAlign = HorizontalAlignment.Right
TextBox1.Font = New Font("Times New Roman", 18)
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
    TextBox1.Text = "1"
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button2.Click
    TextBox1.Text = "2"
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button3.Click
    TextBox1.Text = "3"
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button4.Click
    TextBox1.Text = "4"
End Sub
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button5.Click
    TextBox1.Text = "5"
End Sub

```

```

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button6.Click
    TextBox1.Text = "6"
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button7.Click
    TextBox1.Text = "7"
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button8.Click
    TextBox1.Text = "8"
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button9.Click
    TextBox1.Text = "9"
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button10.Click
    TextBox1.Text = "0"
End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button11.Click
    num2 = TextBox1.Text
    If operators = "+" Then
        TextBox1.Text = num1 + num2
    ElseIf operators = "-" Then
        TextBox1.Text = num1 - num2
    ElseIf operators = "x" Then
        TextBox1.Text = num1 * num2
    ElseIf operators = "/" Then
        TextBox1.Text = num1 / num2
    End If
End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button12.Click
    num1 = TextBox1.Text
    TextBox1.Text = "+"

```

```

        operators = "+"
    End Sub
    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button13.Click
        num1 = TextBox1.Text
        TextBox1.Text = "-"
        operators = "-"
    End Sub
    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button14.Click
        num1 = TextBox1.Text
        TextBox1.Text = "x"
        operators = "x"
    End Sub
    Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button15.Click
        num1 = TextBox1.Text
        TextBox1.Text = "/"
        operators = "/"
    End Sub
    Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button16.Click
        TextBox1.Text = " "
    End Sub
    Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button17.Click
        Me.Close()
    End Sub
End Class

```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 7.2

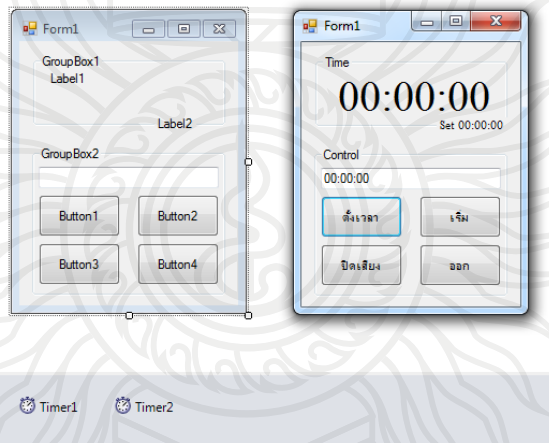
ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 7.2 การแสดงผลโปรแกรมเครื่องคิดเลข

โปรแกรมตั้งเวลาปลุก

ออกแบบสร้างโปรแกรมตั้งเวลาปลุก โดยหน้าจอโปรแกรมมีเวลาปกติ ช่องสำหรับใส่ค่าเวลาที่ต้องการให้มีเสียงเตือน มีปุ่มตั้งเวลา ปุ่มเริ่มทำงาน ปุ่มปิดเสียง และปุ่มออกจากโปรแกรม



ภาพที่ 7.3 โปรแกรมตั้งเวลาปลุก

โปรแกรมคำสั่ง

Public Class Form1

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Label1.TextAlign = HorizontalAlignment.Center
```

```
Label1.Font = New Font("Times New Roman", 30)
```

```
Label1.Text = "00:00:00"
```

```
Label2.TextAlign = HorizontalAlignment.Right
```

```
Label2.Font = New Font("Times New Roman", 8)
```

```
Label2.Text = "Set 00:00:00"
```

```
GroupBox1.Text = "Time"
```

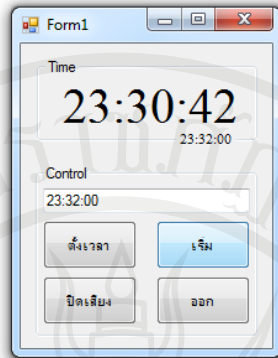
```
GroupBox2.Text = "Control"
```

```

    TextBox1.Text = "00:00:00"
    Button1.Text = "ตั้งเวลา"
    Button2.Text = "เริ่ม"
    Button3.Text = "ปิดเสียง"
    Button4.Text = "ออก"
    Timer1.Start()
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
    Label2.Text = TextBox1.Text
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button2.Click
    Timer2.Start()
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button3.Click
    My.Computer.Audio.Stop()
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button4.Click
    End
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer1.Tick
    Label1.Text = TimeOfDay
End Sub
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer2.Tick
    If Label2.Text = Label1.Text Then
        My.Computer.Audio.Play("C:\animation complete.wav", _
AudioPlayMode.BackgroundLoop)
    End If
End Sub
End Class

```

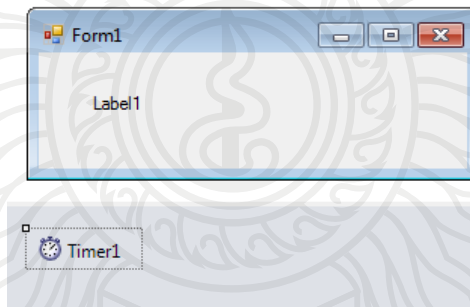
เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 7.4



ภาพที่ 7.4 การแสดงผลโปรแกรมตั้งเวลาปลุก

โปรแกรมแสดงตัวอักษรวิ่ง

ออกแบบโปรแกรมให้มีข้อความวิ่งจากขวาไปซ้าย



ภาพที่ 7.5 โปรแกรมแสดงตัวอักษรวิ่ง

โปรแกรมคำสั่ง

Public Class Form1

Dim x As Integer = 40

Dim y As Integer = 30

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load

Timer1.Start()

Me.Width = 400

Me.Height = 120

Label1.Text = "Hello..."

Label1.Location = New Point(x, y)

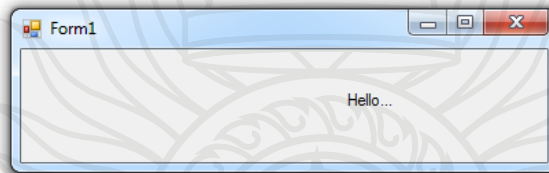
End Sub

```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer1.Tick
    If Label1.Left <= -x Then
        Label1.Left = Me.Width
    Else
        Label1.Left = Label1.Left - 5
    End If
End Sub
End Class

```

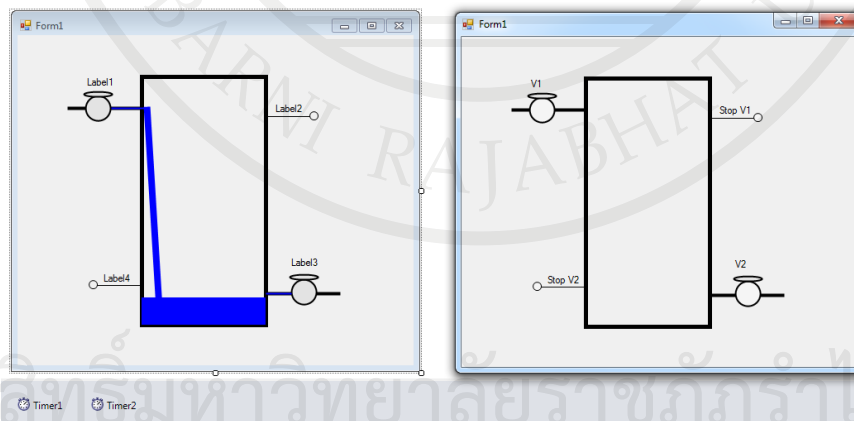
เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 7.6



ภาพที่ 7.6 การแสดงผลโปรแกรมแสดงตัวอักษรวิ่ง

โปรแกรมบอร์ดควบคุมระดับน้ำ

ออกแบบโปรแกรมให้มีหน้าจอบนเป็นรูปแท้งน้ำ มีวาล์วเปิด/ปิด 2 อัน และไฟแสดงสัญญาณเตือนด้านบนและไฟแสดงสัญญาณเตือนด้านล่าง มีการแสดงกราฟฟิคน้ำขึ้นลงตามการเปิด/ปิด โดยเมื่อกดวาล์ว V1 ให้แสดงภาพน้ำเข้าแท้งน้ำและน้ำมีการเพิ่มขึ้นมาเรื่อย ๆ จนถึงไฟแสดงสัญญาณเตือนด้านบนแล้วให้น้ำหยุดเพิ่มพร้อมทั้งปิดวาล์ว V1 ซึ่งหากมีการกดวาล์ว V2 ให้ระดับน้ำลดลงและเมื่อน้ำลดลงมาเรื่อย ๆ จนถึงไฟแสดงสัญญาณเตือนด้านล่างแล้วให้น้ำหยุดไหลพร้อมทั้งปิดวาล์ว V2



ภาพที่ 7.7 โปรแกรมบอร์ดควบคุมระดับน้ำ

โปรแกรมคำสั่ง

Public Class Form1

Dim H As Integer = 0

Dim x As Integer = 150

Dim y As Integer = 350

Dim Status1 As Boolean = True

Dim Status2 As Boolean = True

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

OvalShape1.FillColor = Color.WhiteSmoke

OvalShape2.FillColor = Color.WhiteSmoke

RectangleShape2.FillStyle = PowerPacks.FillStyle.Solid

RectangleShape2.FillColor = Color.Blue

RectangleShape2.Visible = False

OvalShape5.FillStyle = PowerPacks.FillStyle.Solid

OvalShape5.FillColor = Color.WhiteSmoke

OvalShape6.FillStyle = PowerPacks.FillStyle.Solid

OvalShape6.FillColor = Color.WhiteSmoke

LineShape3.Visible = False

LineShape4.Visible = False

LineShape5.Visible = False

RectangleShape2.Height = 38

RectangleShape2.Location = New Point(150, 312)

Label1.Text = "V1"

Label2.Text = "Stop V1"

Label3.Text = "V2"

Label4.Text = "Stop V2"

End Sub

Private Sub OvalShape1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OvalShape1.Click

If Status1 = True Then

OvalShape1.FillColor = Color.Blue

Status1 = False

Timer1.Start()

RectangleShape2.Visible = True

RectangleShape2.Width = 150

RectangleShape2.Height = 0

```

RectangleShape2.Location = New Point(150, 350)
LineShape3.Visible = True
LineShape5.Visible = True
OvalShape6.FillColor = Color.WhiteSmoke
Else
OvalShape1.FillColor = Color.WhiteSmoke
Status1 = True
LineShape3.Visible = False
LineShape5.Visible = False
Timer1.Stop()
End If
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer1.Tick
H = H + 3
y = y - 3
RectangleShape2.Height = H
RectangleShape2.Location = New Point(150, y)
If H >= 250 Then
Timer1.Stop()
OvalShape1.FillColor = Color.WhiteSmoke
Status1 = True
LineShape3.Visible = False
LineShape4.Visible = False
LineShape5.Visible = False
OvalShape5.FillColor = Color.Red
End If
End Sub
Private Sub OvalShape2_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles OvalShape2.Click
If Status2 = True Then
OvalShape2.FillColor = Color.Blue
Status2 = False
OvalShape5.FillColor = Color.WhiteSmoke
LineShape4.Visible = True
Timer2.Start()
Else

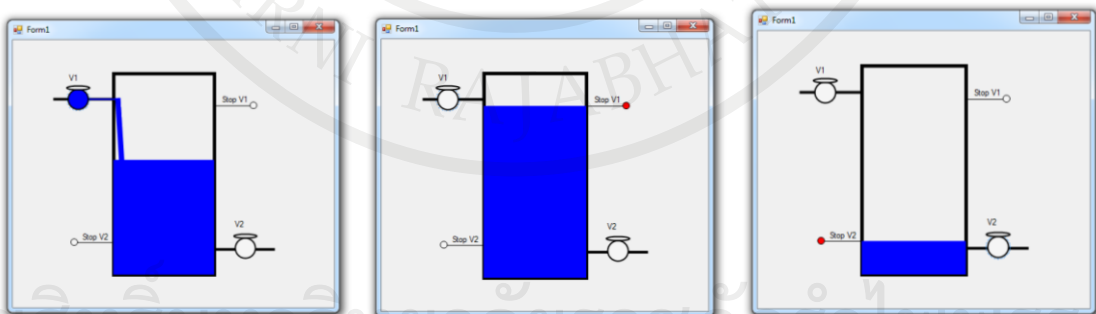
```

```

OvalShape2.FillColor = Color.WhiteSmoke
Status2 = True
LineShape4.Visible = False
Timer2.Stop()
End If
End Sub
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer2.Tick
    H = H - 3
    y = y + 3
    RectangleShape2.Height = H
    RectangleShape2.Location = New Point(150, y)
    If H <= 50 Then
        Timer2.Stop()
        OvalShape1.FillColor = Color.WhiteSmoke
        Status2 = True
        LineShape3.Visible = False
        LineShape4.Visible = False
        LineShape5.Visible = False
        OvalShape6.FillColor = Color.Red
        OvalShape2.FillColor = Color.WhiteSmoke
    End If
End Sub
End Class

```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 7.8

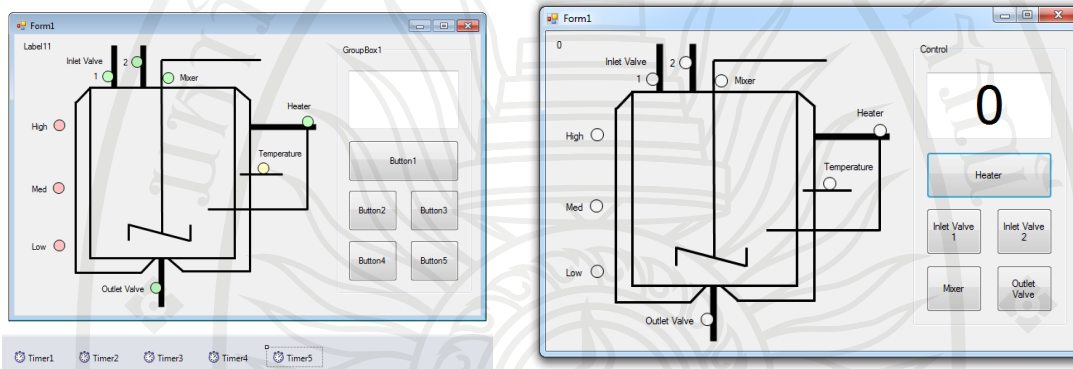


ภาพที่ 7.8 การแสดงผลโปรแกรมบอร์ดควบคุมระดับน้ำ

โปรแกรมบอร์ดควบคุมไฟในถังผสม

ออกแบบบอร์ดควบคุมไฟในถังผสมดังนี้

1. สามารถกดการแสดงผลการควบคุมไฟในถังผสมที่ปุ่ม Mixer ได้
2. เมื่อกด Heater จะแสดงผลอุณหภูมิที่วิ่งขึ้นไปจนถึง 80 แล้วหยุดพร้อมทั้งไฟ Temperature ติด
3. เมื่อกด Inlet Valve 1 ไฟแสดงระดับน้ำจะค่อย ๆ ติดและถ้ากด Inlet Valve 2 ด้วยไฟแสดงระดับน้ำจะติดไวขึ้นเหมือนกับลักษณะการเปิดน้ำเข้า 1 วาล์ว กับเปิดน้ำเข้า 2 วาล์ว
4. เมื่อกด Outlet Valve ให้ไฟแสดงระดับน้ำค่อย ๆ ลดลง



ภาพที่ 7.9 โปรแกรมบอร์ดควบคุมไฟในถังผสม

โปรแกรมคำสั่ง

Public Class Form1

Dim t1 As Integer

Dim t2 As Integer

Dim t3 As Integer

Dim Status1 As Boolean = True

Dim Status2 As Boolean = True

Dim Status3 As Boolean = True

Dim Status4 As Boolean = True

Dim Status5 As Boolean = True

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _

System.EventArgs) Handles MyBase.Load

GroupBox1.Text = "Control"

Button1.Text = "Heater"

Button2.Text = "Inlet Valve 1"

Button3.Text = "Inlet Valve 2"

Button4.Text = "Mixer"


```

Button5.Text = "Outlet Valve"
TextBox1.Text = "0"
TextBox1.TextAlign = HorizontalAlignment.Center
OvalShape1.FillColor = Color.WhiteSmoke
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.WhiteSmoke
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.WhiteSmoke
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.WhiteSmoke
Timer1.Stop()
Label11.Text = "0"
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
    If Status1 = True Then
        Timer1.Start()
        OvalShape1.FillColor = Color.GreenYellow
        Button1.Text = "Stop Heater"
        Status1 = False
    Else
        Timer1.Stop()
        OvalShape1.FillColor = Color.WhiteSmoke
        Button1.Text = "Start Heater"
        Status1 = True
    End If
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer1.Tick
    TextBox1.Text = t1
    t1 = t1 + 1
    If t1 > 80 Then
        Timer1.Stop()
        OvalShape9.FillColor = Color.Yellow
        OvalShape1.FillColor = Color.WhiteSmoke

```

```

    t1 = 0
End If
End Sub
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles TextBox1.TextChanged
    t1 = TextBox1.Text
    If Status1 = False Then
        If t1 <= 80 Then
            Timer1.Start()
            OvalShape9.FillColor = Color.WhiteSmoke
            OvalShape1.FillColor = Color.GreenYellow
        Else
            Timer1.Stop()
            OvalShape1.FillColor = Color.WhiteSmoke
        End If
    End If
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button2.Click
    If Status2 = True Then
        OvalShape2.FillColor = Color.GreenYellow
        Timer2.Start()
        Timer4.Start()
        Status2 = False
    Else
        OvalShape2.FillColor = Color.WhiteSmoke
        Timer2.Stop()
        Timer4.Stop()
        Status2 = True
    End If
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button3.Click
    If Status3 = True Then
        OvalShape3.FillColor = Color.GreenYellow
        Timer3.Start()
        Timer4.Start()

```

```

    Status3 = False
Else
    OvalShape3.FillColor = Color.WhiteSmoke
    Timer3.Stop()
    Timer4.Stop()
    Status3 = True
End If
End Sub
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer2.Tick
    t2 = t2 + 1
    Label11.Text = t2
End Sub
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer3.Tick
    t2 = t2 + 1
    Label11.Text = t2
End Sub
Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer4.Tick
    If Label11.Text >= 210 Then
        OvalShape6.FillColor = Color.Red
        Timer2.Stop()
        Timer3.Stop()
        Timer4.Stop()
        Status2 = True
        Status3 = True
        OvalShape2.FillColor = Color.WhiteSmoke
        OvalShape3.FillColor = Color.WhiteSmoke
    ElseIf Label11.Text >= 110 Then
        OvalShape7.FillColor = Color.Red
    ElseIf Label11.Text >= 10 Then
        OvalShape8.FillColor = Color.Red
    End If
End Sub

```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button4.Click
```

```
    If Status4 = True Then
```

```
        OvalShape5.FillColor = Color.GreenYellow
```

```
        Status4 = False
```

```
    Else
```

```
        OvalShape5.FillColor = Color.WhiteSmoke
```

```
        Status4 = True
```

```
    End If
```

```
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button5.Click
```

```
    If Status5 = True Then
```

```
        OvalShape4.FillColor = Color.GreenYellow
```

```
        Timer5.Start()
```

```
        Status5 = False
```

```
    Else
```

```
        OvalShape4.FillColor = Color.WhiteSmoke
```

```
        Timer5.Stop()
```

```
        Status5 = True
```

```
    End If
```

```
End Sub
```

```
Private Sub Timer5_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer5.Tick
```

```
    t2 = t2 - 1
```

```
    Label11.Text = t2
```

```
    If Label11.Text <= 9 Then
```

```
        OvalShape8.FillColor = Color.WhiteSmoke
```

```
    ElseIf Label11.Text <= 109 Then
```

```
        OvalShape7.FillColor = Color.WhiteSmoke
```

```
    ElseIf Label11.Text <= 209 Then
```

```
        OvalShape6.FillColor = Color.WhiteSmoke
```

```
    End If
```

```
    If Label11.Text <= 0 Then
```

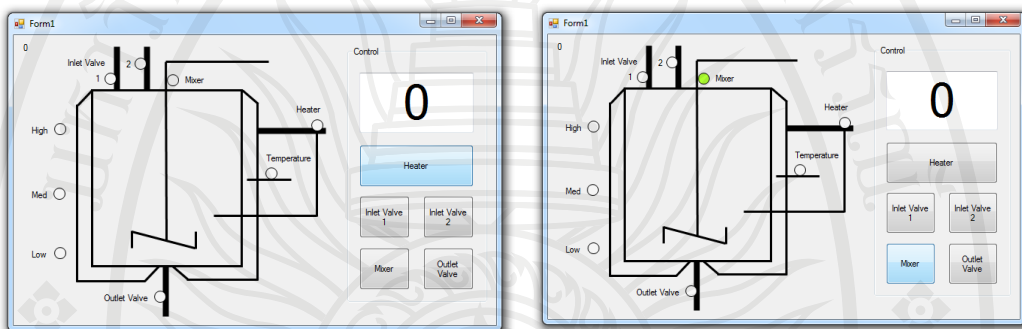
```
        Timer5.Stop()
```

```
        OvalShape4.FillColor = Color.WhiteSmoke
```

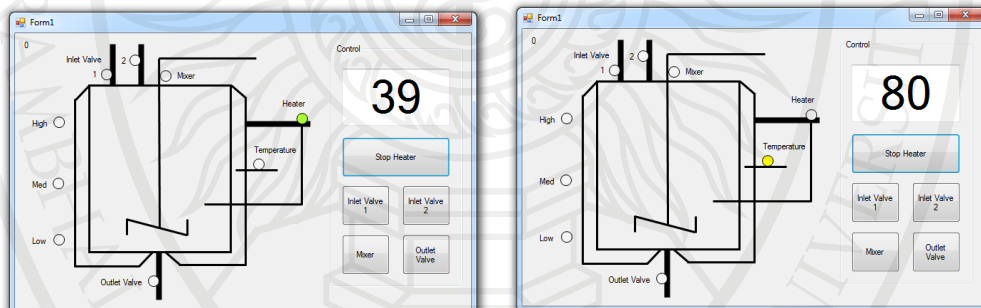
```
        OvalShape9.FillColor = Color.WhiteSmoke
```

End If
 End Sub
 End Class

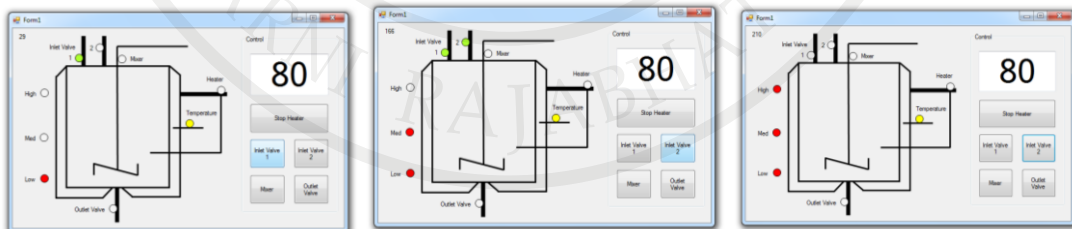
เมื่อกด F5 จะได้ผลลัพธ์ สามารถกดการแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมที่ปั๊ม Mixer ดังภาพที่ 7.10 ,การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมที่ปั๊ม Heater ดังภาพที่ 7.11 , การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมเมื่อกด Inlet Valve ดังภาพที่ 7.12 และการแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมเมื่อกด Outlet Valve ดังภาพที่ 7.13



ภาพที่ 7.10 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมที่ปั๊ม Mixer

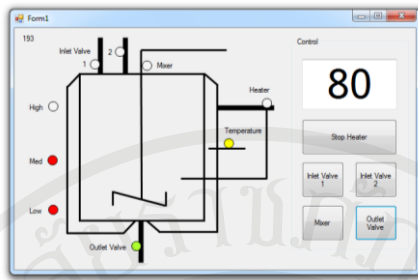


ภาพที่ 7.11 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมที่ปั๊ม Heater



ภาพที่ 7.12 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมเมื่อกดปั๊ม Inlet Valve

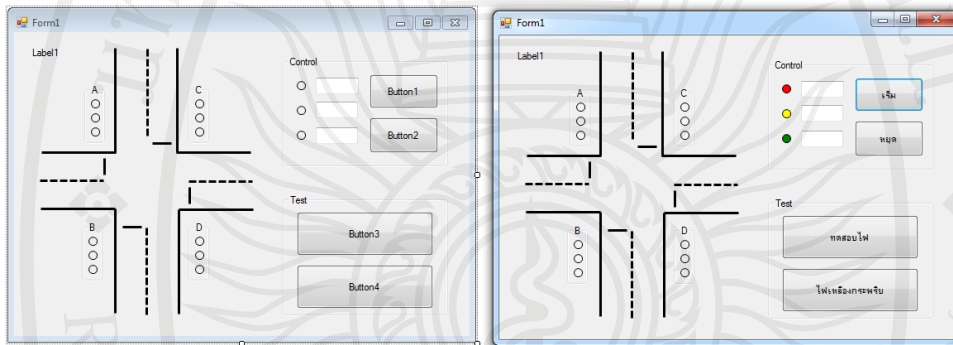
ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี



ภาพที่ 7.13 การแสดงผลโปรแกรมบอร์ดควบคุมไฟในถังผสมเมื่อกดปุ่ม Outlet Valve

โปรแกรมควบคุมไฟจราจร

ออกแบบโปรแกรมไฟจราจร โดยสามารถควบคุมความนานของไฟเขียว เหลือง แดงได้ตลอดจนมีปุ่มทดสอบไฟและปุ่มให้ไฟเหลืองกระพริบไปเรื่อย ๆ



ภาพที่ 7.14 โปรแกรมควบคุมไฟจราจร

โปรแกรมคำสั่ง

Public Class Form1

Dim T As Integer

Dim C As Integer

Dim R As Integer

Dim Y As Integer

Dim G As Integer

Dim AG As Integer

Dim AY As Integer

Dim BG As Integer

Dim BY As Integer

Dim CG As Integer

Dim CY As Integer

Dim DG As Integer

Dim DY As Integer

Dim Status1 As Boolean = True

Dim Status2 As Boolean = True

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Button1.Text = "เริ่ม"

Button2.Text = "หยุด"

Button3.Text = "ทดสอบไฟ"

Button4.Text = "ไฟเหลืองกระพริบ"

OvalShape1.FillStyle = PowerPacks.FillStyle.Solid

OvalShape2.FillStyle = PowerPacks.FillStyle.Solid

OvalShape3.FillStyle = PowerPacks.FillStyle.Solid

OvalShape4.FillStyle = PowerPacks.FillStyle.Solid

OvalShape5.FillStyle = PowerPacks.FillStyle.Solid

OvalShape6.FillStyle = PowerPacks.FillStyle.Solid

OvalShape7.FillStyle = PowerPacks.FillStyle.Solid

OvalShape8.FillStyle = PowerPacks.FillStyle.Solid

OvalShape9.FillStyle = PowerPacks.FillStyle.Solid

OvalShape10.FillStyle = PowerPacks.FillStyle.Solid

OvalShape11.FillStyle = PowerPacks.FillStyle.Solid

OvalShape12.FillStyle = PowerPacks.FillStyle.Solid

OvalShape13.FillStyle = PowerPacks.FillStyle.Solid

OvalShape14.FillStyle = PowerPacks.FillStyle.Solid

OvalShape15.FillStyle = PowerPacks.FillStyle.Solid

OvalShape1.FillColor = Color.WhiteSmoke

OvalShape2.FillColor = Color.WhiteSmoke

OvalShape3.FillColor = Color.WhiteSmoke

OvalShape4.FillColor = Color.WhiteSmoke

OvalShape5.FillColor = Color.WhiteSmoke

OvalShape6.FillColor = Color.WhiteSmoke

OvalShape7.FillColor = Color.WhiteSmoke

OvalShape8.FillColor = Color.WhiteSmoke

OvalShape9.FillColor = Color.WhiteSmoke

OvalShape10.FillColor = Color.WhiteSmoke

OvalShape11.FillColor = Color.WhiteSmoke

OvalShape12.FillColor = Color.WhiteSmoke

OvalShape13.FillColor = Color.Red

OvalShape14.FillColor = Color.Yellow

```
OvalShape15.FillColor = Color.Green
```

```
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles Timer1.Tick
```

```
T = T + 1
```

```
Label1.Text = T
```

```
Select Case T
```

```
Case AG
```

```
OvalShape1.FillColor = Color.WhiteSmoke
```

```
OvalShape2.FillColor = Color.WhiteSmoke
```

```
OvalShape3.FillColor = Color.Green
```

```
OvalShape4.FillColor = Color.Red
```

```
OvalShape5.FillColor = Color.WhiteSmoke
```

```
OvalShape6.FillColor = Color.WhiteSmoke
```

```
OvalShape7.FillColor = Color.Red
```

```
OvalShape8.FillColor = Color.WhiteSmoke
```

```
OvalShape9.FillColor = Color.WhiteSmoke
```

```
OvalShape10.FillColor = Color.Red
```

```
OvalShape11.FillColor = Color.WhiteSmoke
```

```
OvalShape12.FillColor = Color.WhiteSmoke
```

```
Case AY
```

```
OvalShape1.FillColor = Color.WhiteSmoke
```

```
OvalShape2.FillColor = Color.Yellow
```

```
OvalShape3.FillColor = Color.WhiteSmoke
```

```
OvalShape4.FillColor = Color.Red
```

```
OvalShape5.FillColor = Color.WhiteSmoke
```

```
OvalShape6.FillColor = Color.WhiteSmoke
```

```
OvalShape7.FillColor = Color.Red
```

```
OvalShape8.FillColor = Color.WhiteSmoke
```

```
OvalShape9.FillColor = Color.WhiteSmoke
```

```
OvalShape10.FillColor = Color.Red
```

```
OvalShape11.FillColor = Color.WhiteSmoke
```

```
OvalShape12.FillColor = Color.WhiteSmoke
```

```
Case BG
```

```
OvalShape1.FillColor = Color.Red
```

```
OvalShape2.FillColor = Color.WhiteSmoke
```

```
OvalShape3.FillColor = Color.WhiteSmoke
```


OvalShape4.FillColor = Color.WhiteSmoke
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.Green
OvalShape7.FillColor = Color.Red
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.WhiteSmoke
OvalShape10.FillColor = Color.Red
OvalShape11.FillColor = Color.WhiteSmoke
OvalShape12.FillColor = Color.WhiteSmoke

Case BY

OvalShape1.FillColor = Color.Red
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.WhiteSmoke
OvalShape5.FillColor = Color.Yellow
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.Red
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.WhiteSmoke
OvalShape10.FillColor = Color.Red
OvalShape11.FillColor = Color.WhiteSmoke
OvalShape12.FillColor = Color.WhiteSmoke

Case CG

OvalShape1.FillColor = Color.Red
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.Red
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.WhiteSmoke
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.Green
OvalShape10.FillColor = Color.Red
OvalShape11.FillColor = Color.WhiteSmoke
OvalShape12.FillColor = Color.WhiteSmoke

Case CY

OvalShape1.FillColor = Color.Red
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.Red
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.WhiteSmoke
OvalShape8.FillColor = Color.Yellow
OvalShape9.FillColor = Color.WhiteSmoke
OvalShape10.FillColor = Color.Red
OvalShape11.FillColor = Color.WhiteSmoke
OvalShape12.FillColor = Color.WhiteSmoke

Case DG

OvalShape1.FillColor = Color.Red
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.Red
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.Red
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.WhiteSmoke
OvalShape10.FillColor = Color.WhiteSmoke
OvalShape11.FillColor = Color.WhiteSmoke
OvalShape12.FillColor = Color.Green

Case DY

OvalShape1.FillColor = Color.Red
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.Red
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.Red
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.WhiteSmoke

```

        OvalShape10.FillColor = Color.WhiteSmoke
        OvalShape11.FillColor = Color.Yellow
        OvalShape12.FillColor = Color.WhiteSmoke
    End Select
    If T = DY + Y Then
        T = 0
    End If
End Sub
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Timer2.Tick
    C = C + 1
    Label1.Text = C
    If C <= 10 Then
        OvalShape1.FillColor = Color.WhiteSmoke
        OvalShape2.FillColor = Color.Yellow
        OvalShape3.FillColor = Color.WhiteSmoke
        OvalShape4.FillColor = Color.WhiteSmoke
        OvalShape5.FillColor = Color.Yellow
        OvalShape6.FillColor = Color.WhiteSmoke
        OvalShape7.FillColor = Color.WhiteSmoke
        OvalShape8.FillColor = Color.Yellow
        OvalShape9.FillColor = Color.WhiteSmoke
        OvalShape10.FillColor = Color.WhiteSmoke
        OvalShape11.FillColor = Color.Yellow
        OvalShape12.FillColor = Color.WhiteSmoke
    Else
        OvalShape1.FillColor = Color.WhiteSmoke
        OvalShape2.FillColor = Color.WhiteSmoke
        OvalShape3.FillColor = Color.WhiteSmoke
        OvalShape4.FillColor = Color.WhiteSmoke
        OvalShape5.FillColor = Color.WhiteSmoke
        OvalShape6.FillColor = Color.WhiteSmoke
        OvalShape7.FillColor = Color.WhiteSmoke
        OvalShape8.FillColor = Color.WhiteSmoke
        OvalShape9.FillColor = Color.WhiteSmoke
        OvalShape10.FillColor = Color.WhiteSmoke
        OvalShape11.FillColor = Color.WhiteSmoke
    End If
End Sub

```

```

    OvalShape12.FillColor = Color.WhiteSmoke
End If
If C >= 15 Then C = 0
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
    Timer1.Start()
    R = TextBox1.Text
    Y = TextBox2.Text
    G = TextBox3.Text
    AG = 1
    AY = G
    BG = AY + Y
    BY = BG + G
    CG = BY + Y
    CY = CG + G
    DG = CY + Y
    DY = DG + G
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button2.Click
    Timer1.Stop()
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button3.Click
    If Status1 = True Then
        OvalShape1.FillColor = Color.Red
        OvalShape2.FillColor = Color.Yellow
        OvalShape3.FillColor = Color.Green
        OvalShape4.FillColor = Color.Red
        OvalShape5.FillColor = Color.Yellow
        OvalShape6.FillColor = Color.Green
        OvalShape7.FillColor = Color.Red
        OvalShape8.FillColor = Color.Yellow
        OvalShape9.FillColor = Color.Green
        OvalShape10.FillColor = Color.Red
        OvalShape11.FillColor = Color.Yellow

```

```

OvalShape12.FillColor = Color.Green
Status1 = False
Else
OvalShape1.FillColor = Color.WhiteSmoke
OvalShape2.FillColor = Color.WhiteSmoke
OvalShape3.FillColor = Color.WhiteSmoke
OvalShape4.FillColor = Color.WhiteSmoke
OvalShape5.FillColor = Color.WhiteSmoke
OvalShape6.FillColor = Color.WhiteSmoke
OvalShape7.FillColor = Color.WhiteSmoke
OvalShape8.FillColor = Color.WhiteSmoke
OvalShape9.FillColor = Color.WhiteSmoke
OvalShape10.FillColor = Color.WhiteSmoke
OvalShape11.FillColor = Color.WhiteSmoke
OvalShape12.FillColor = Color.WhiteSmoke
Status1 = True
End If
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Button4.Click
If Status2 = True Then
Button4.Text = "เริ่มทดสอบ ไฟเหลืองกระพริบ"
Status2 = False
Timer2.Start()
Else
Button4.Text = "หยุด ไฟเหลืองกระพริบ"
Status2 = True
Timer2.Stop()
End If
End Sub
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles TextBox1.TextChanged
TextBox3.Text = TextBox1.Text
End Sub

```

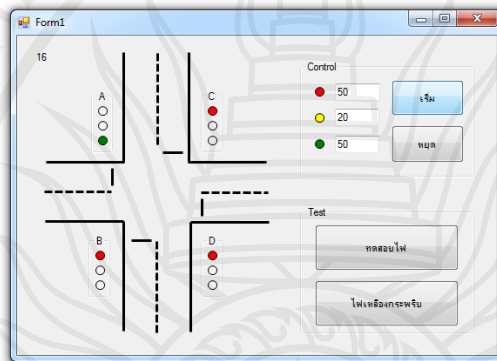
```
Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles TextBox3.TextChanged
```

```
    TextBox1.Text = TextBox3.Text
```

```
End Sub
```

```
End Class
```

เมื่อกด F5 จะได้ผลลัพธ์ ดังภาพที่ 7.15



ภาพที่ 7.15 การแสดงผลโปรแกรมควบคุมไฟจราจร

สรุป

โจทย์โปรแกรมสำหรับศึกษาเรียนรู้ได้แก่โปรแกรมเครื่องคิดเลข, โปรแกรมตั้งเวลาปลุก, โปรแกรมแสดงอักษรวิ่ง, โปรแกรมบอร์ดควบคุมระดับน้ำ, โปรแกรมบอร์ดควบคุมไฟในถังผสมและโปรแกรมควบคุมไฟจราจร สามารถเป็นตัวช่วยให้ผู้ที่สนใจได้นำไปใช้ออกแบบและพัฒนาสำหรับงานวิศวกรรมต่อไปได้ ทั้งนี้ นักศึกษาสามารถหาข้อมูลตัวอย่างวิธีการเขียนโปรแกรมคำสั่งและการใช้คำสั่งเพิ่มเติมได้ที่ Microsoft Visual Basic 2010 Step by Step. (Halvorson, M. 2010). และ Microsoft Visual Basic 2010 Developer's Handbook, (Loffelmann, K., & CallaPurohit, S. 2010).

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

แบบฝึกหัดบทที่ 7

1. จากโปรแกรมเครื่องคิดเลข หากต้องการเพิ่มปุ่มยกกำลังต้องเขียนโปรแกรมคำสั่งอย่างไร
2. จากโปรแกรมเครื่องคิดเลข หากต้องการให้ช่องแสดงผลการคำนวณมีขนาดใหญ่ขึ้นต้องปรับค่าหรือเขียนโปรแกรมคำสั่งอย่างไร
3. จงออกแบบโปรแกรมตั้งเวลาปลุกสำหรับเด็ก
4. จงออกแบบโปรแกรมตั้งเวลาสำหรับให้อาหารสุนัข โดยเมื่อถึงเวลาที่ตั้งให้แสดงสัญญาณไฟเตือน
5. จากโปรแกรมแสดงตัวอักษรวิ่ง หากต้องการให้ข้อความมีขนาดใหญ่ขึ้นและมีการเปลี่ยนสีของตัวอักษรไปเรื่อย ๆ ต้องเขียนโปรแกรมคำสั่งอย่างไร
6. จากโปรแกรมบอร์ดควบคุมระดับน้ำ หากต้องการให้ระดับน้ำเพิ่ม/ลดไวขึ้นต้องปรับ โปรแกรมคำสั่งอย่างไร
7. จงคิดอัลกอริทึมและเขียนโปรแกรมเพื่อปรับโปรแกรมบอร์ดควบคุมไฟในถังผสมใหม่ โดยให้บอร์ดควบคุมไฟนี้ทำงานอัตโนมัติ หลังจากกดปุ่ม Start ให้ Inlet Valve 1 ทำงาน
8. จงเขียนผังงานโปรแกรมควบคุมไฟจราจร
9. จงออกแบบโปรแกรมสำหรับงานวิศวกรรมมา 1 เรื่อง

เอกสารอ้างอิง

- ธีรวัฒน์ ประกอบผล, (2554). คู่มือการเขียนแอปพลิเคชันด้วย Visual Basic 2010. กรุงเทพมหานคร: ชิมพลีฟาย.
- สัจจะ จรัสรุ่งรวีวร, (2554). คู่มือเรียนและใช้งาน Visual Basic 2010. นนทบุรี: ไอทีซี พีริเมียร์จำกัด.
- Halvorson, M. (2010). Microsoft Visual Basic 2010 Step by Step. United States of America.
- Loffelmann, K., & CallaPurohit ,S.(2010). Microsoft Visual Basic 2010 Developer's Handbook, Canada: O' Reilly Media.

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี

บรรณานุกรม

- เกียรติศักดิ์ ทรงแก้ว. (2552). ความหมายและหลักการทำงานของ SDRAM และ DDR. [On-line] Available : <http://www.vcharkarn.com/blog/40889>. [2556, ตุลาคม 20].
- จักรกรฤษณ์ แก้วไพฑูรย์. (ม.ป.ป.). Intel core i7 โพรเซสเซอร์สุดยอแห่งยุค [On-line] Available : <http://www.chandra.ac.th/office/ict/project/3G%20Technology/Content6.php>. [2556, ตุลาคม 20].
- ณัฐกร สงคราม. (ม.ป.ป.). บทเรียนโปรแกรมการเรียนการสอนผ่านเว็บเรื่องคอมพิวเตอร์เบื้องต้น . [On-line] Available : http://www.thaiwbi.com/course/Intro_com/Intro_com/wbi1/hie/homepage.htm. [2556, ตุลาคม 6].
- ณรงค์ นันทวิจิตร. (2547). Mainboard. [On-line] Available : <http://www.brr.ac.th/oldweb/knowledge/hardware/mainboard/main.htm>. [2556, ตุลาคม 20].
- ธีรวัฒน์ ประกอบผล, (2554). คู่มือการเขียนแอปพลิเคชันด้วย Visual Basic 2010. กรุงเทพมหานคร: ชิมพลีฟาย.
- ธีรเดช จันทร์วิเมลีอง.(ม.ป.ป.). ประวัติภาษา BASIC (BASIC Programming Language History). [On-line] Available : <http://www.vb.in.th/2007/06/basic-basic-programming-language.html> [2556, ตุลาคม 6].
- นัฐพงศ์ สงเนียม, (ม.ป.ป.). การเขียนโปรแกรมด้วย Microsoft Visual Studio 2010. [On-line] Available : http://www.siam2dev.com/e_learning/algorithm/lec04_vb_net2010_variables_declaration.pdf. [2556, December 8].
- นिरนาม, (ม.ป.ป.). Visual basic 2010 gramming. [On-line] Available : <http://visualgramming.blogspot.com/2013/11/reserved-words.html#> [2556, December 8].
- นिरนาม (2555) จอภาพหรือมอนิเตอร์ (Monitor) CRT/LCD/LED คืออะไร. [On-line] Available : <http://www.thaiwebsocial.com>. [2556, ตุลาคม 20].
- นिरนาม (2556). สัญญาณอันตราย ฮาร์ดดิสก์ใกล้พัง. [On-line] Available : <http://www.rangforever.com/news.php?id=61>. [2556, ตุลาคม 20].
- มงคล มะเมียทอง. (ม.ป.ป.). เอกสารประกอบการสอน วิชาการเขียนโปรแกรมขั้นสูง ระดับชั้น ม.6. [On-line] Available : http://www.ro.ac.th/mongkolro/doc/setup_vs2010.pdf. [2556, ตุลาคม 6].
- ยงยุทธ ชมไชย. (ม.ป.ป.). หลักการทำงานของคอมพิวเตอร์. [On-line] Available : <https://sites.google.com/site/kruyutswb/3-2-hlak-kar-thangan-khxmphiwtextr>. [2556, ตุลาคม 6].
- ลักษณะ วาทิน. (ม.ป.ป.). ประวัติความเป็นมาของ VisualBasic. [On-line] Available : <https://sites.google.com/site/luxsanavathin/gui/chnid-khxm-laea-tawpaer>. [2556, ตุลาคม 6].

บรรณานุกรม (ต่อ)

- วิโรจน์ ชัยมูลและสุพรรณษา ยวงทอง. (2552). ความรู้เบื้องต้นเกี่ยวกับคอมพิวเตอร์และเทคโนโลยีสารสนเทศ. กรุงเทพมหานคร: โปรวิชั่น.
- สัจจะ จรัสรุ่งรวีวร, (2554). คู่มือเรียนและใช้งาน Visual Basic 2010. นนทบุรี: ไอทีซี พรีเมียร์จำกัด.
- สุรัตน์ โคอินทรางกูร, (ม.ป.ป.). **การเขียน Flowchart เบื้องต้น**. [On-line] Available : <http://www.bus.tu.ac.th/usr/surat/mis201/homepage/cmat/sheet/flowchar.doc>. [2556, ตุลาคม 27].
- ศุภกรณา ไกรโนนทอง. (ม.ป.ป.). กลุ่มคีย์ 5 กลุ่มบนคีย์บอร์ดมาตรฐาน. [On-line] Available : <https://sites.google.com/site/sooknapa235/keyboard> [2556, ตุลาคม 6].
- โอภาส เอี่ยมสิริวงศ์. (2547). วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น.
- Alibaba.com. (n.d.). Small Black USB Wired Optical Light Scroll Wheel Mice Mouse for PC Laptop Computer. [On-line] Available : http://www.alibaba.com/product-detail/Small-Black-USB-Wired-Optical-Light_1978012783.html. [2556, October 6].
- AltoEdge Pty. (n.d.). Microphones. [On-line] Available : <http://www.altoedge.com/microphones/>. [2556, October 6].
- Amazon.com. (n.d.). HP Wireless Elite Keyboard v2. [On-line] Available : <http://www.amazon.com/HP-Wireless-Elite-Keyboard-v2/dp/B005HNHRA6>. [2556, October 6].
- Amazon.com. (n.d.). PalmOne Portable Keyboard for Palm m100/m105, III Series, and VII Series Handhelds. [On-line] Available : <http://www.amazon.com/PalmOne-Portable-Keyboard-Palm-Handhelds/dp/B00004WHF9>. [2556, October 6].
- Amazon.com. (n.d.). Logitech Cordless Optical TrackMan. [On-line] Available : http://www.amazon.com/Logitech-Cordless-Optical-TrackMan/dp/B0032KYA6K/ref=sr_1_1?s=pc&ie=UTF8&qid=1461249934&sr=1-1 . [2556, October 6].
- Barcode118.com (2014). Barcode Scanning. [On-line] Available : <http://barcode118.com/barcode-scanning.html>. [2556, October 20].
- Bio office. (n.d.). D200 DeskTop Series FingerPrint Employee Time Clock. [On-line] Available : <http://store.bio-office.com>. [2556, October 20].
- Bluhomes (n.d.). A better client experience. [On-line] Available : <https://www.bluhomes.com/our-vision> [2556, November 24].
- Chatsworth (2010). ACP-2200 OMR. [On-line] Available : http://www.chatsworthdata.com/optical_mark_readers-acp_2200omr.html. [2556, October 20].
- CircuitsToday. (n.d.). Different Types Of Digital Cameras. [On-line] Available : <http://www.circuitstoday.com/different-types-of-digital-cameras>. [2556, October 6].
- Dell Inc. (n.d.). Electronic, Accessories, and Software Categories. [On-line] Available : <http://accessories.ap.dell.com/sna/default.aspx>. [2556, October 6].

บรรณานุกรม (ต่อ)

- DriverBasket (n.d.). Samsung SCX-3405FW Wireless Multifunction Printer Driver. [On-line] Available : <http://driverbasket.com/samsung-scx-3405fw-printer-drivers-download/>. [2556, October 20].
- EncomPos. (n.d.). Receipt Printers. [On-line] Available : <http://encomposoftware.com/hardware.html>. [2556, October 20].
- Erik D. Kennedy (n.d.). 7 Rules for Creating Gorgeous UI part 1. [On-line]. <https://medium.com/@erikdkennedy/7-rules-for-creating-gorgeous-ui-part-1-559d4e805cda#8v06ghkou>. [2556, November 10].
- Erik D. Kennedy (n.d.). 7 Rules for Creating Gorgeous UI part 2. [On-line]. <https://medium.com/@erikdkennedy/7-rules-for-creating-gorgeous-ui-part-2-430de537ba96#ypo0j45u8>. [2556, November 10].
- Garmin (n.d.). On the Road nuvi42 [On-line] Available : <http://www.garmin.co.th/products/ontheroad/nuvi42/>[2556, December 8].
- Google Sites. (n.d.). Printer and Scanner. [On-line] Available : <https://sites.google.com/site/printerandscanner/>. [2556, October 6].
- Halvorson, M. (2010). Microsoft Visual Basic 2010 Step by Step. United States of America.
- Hearbeats (n.d.). beats-studio-red. [On-line] Available : <http://www.hearbeats.com/product/14/%E0%B8%AB%E0%B8%B9%E0%B8%9F%E0%B8%B1%E0%B8%87-beats-studio-red>. [2556, October 20].
- Huge (n.d.). A new home for Google's universal design language, Material Design, that brings the new spec to life on any size screen. [On-line] Available : <http://www.hugeinc.com/case-study/material-design/>[2556, November 24].
- iPhone J.D. (2011). Review: Kensington Virtuoso Stylus for Tablet -- full length stylus for the iPad. [On-line] Available : http://www.iphonejd.com/iphone_jd/2011/08/review-kensington-virtuoso-stylus-for-tablet-full-length-stylus-for-the-ipad.html. [2556, October 6].
- Kerem (n.d.). Kerem Suer designs systematic methods to help humans interact with machines. [On-line] Available : <http://kerem.co/>. [2556, November 11].
- Loffelmann, K.,& CallaPurohit ,S.(2010). Microsoft Visual Basic 2010 Developer's Handbook, Canada: O' Reilly Media.
- Media Data Systems Pte Ltd. (n.d.). Portable Digital Video Camera with 5.0-megapixel CMOS Sensor, Supports USB 2.0 PC Interface. [On-line] Available : <http://www.globalsources.com/gsol/Digital-video/p/sm/1042430107.htm#1042430107>. [2556, October 6].

บรรณานุกรม (ต่อ)

- Medium (n.d.). Floor Fade [On-line] Available : <https://medium.com/>[2556, November 24].
- Miguel Oliva Márquez (n.d.). Modern Honolulu [On-line] Available : <http://www.designil.com/miguelolivamarquez.com> [2556, November 24].
- Notebookspec (2011). Printer. [On-line] Available : <http://notebookspec.com/web/wp-content/uploads/2011/11/L100-550-x-3101.png>. [2556, October 20].
- OfficeMate (n.d.). Printer. [On-line] Available : <http://store.officemate.co.th/images/zpimage/4550008.jpg>. [2556, October 20].
- Pallab Chatterjee (2012). Embedded Mobile Driven by Human Interface. [On-line] Available : <http://www.digikey.com/en/articles/techzone/2012/feb/embedded-mobile-driven-by-human-interface>. [2556, October 20].
- Pelaajan Valinta. (2009). PC Pro Flight 2 joystick (PC). [On-line] Available : <http://www.pelaajanvalinta.fi/PC-Pro-Flight-2-joystick-PC>. [2556, October 6].
- Perfect Solution (n.d.). ระบบร้านอาหาร โปรแกรมร้านอาหาร POS ร้านอาหาร รองรับการใช้งานด้วย iPod, iPad, iPhone ใช้งานง่าย [On-line] Available : <http://pos-restaurantprogram.blogspot.com/2013/06/ipad-take-order.html>[2556, December 8].
- Peter Vogel & Kim Ryrie. (1979). Fairlight Computer Music Instrument. [On-line] Available : <http://120years.net/fairlight-computer-music-instrument-peter-vogel-kim-ryrie-australia-1979/>. [2556, October 6].
- Piotr Kwiatkowski (n.d.). Music UI – UI Design Concept [On-line] Available : <http://www.piotrkwiatkowski.co.uk/>[2556, November 11].
- PortableApps.com. (2012). On-Screen Keyboard Portable. [On-line] Available : http://portableapps.com/apps/accessibility/on-screen_keyboard_portable. [2556, October 6].
- Redmond Pie. (n.d.). Future Notebooks Will Have Fingerprint Scanners Built Right Into The Touchpad. [On-line] Available : <http://www.redmondpie.com/future-notebooks-will-have-fingerprint-scanners-built-right-into-the-touchpad/>. [2556, October 6].
- Sajee (2013). speaker. [On-line] Available : <https://sajee1234.files.wordpress.com/2013/07/81934.jpg>. [2556, October 20].
- SlashGee. (n.d.). LG ET83 touchscreen monitor wants your Windows 8 fingers. [On-line] Available : <http://www.slashgear.com/lg-et83-touchscreen-monitor-wants-your-windows-8-fingers-31254980/>. [2556, October 6].

บรรณานุกรม (ต่อ)

- Smashing Magazine (2010). Color Theory For Designers: Creating Your Own Color Palettes. [On-line] Available : <https://www.smashingmagazine.com/2010/02/color-theory-for-designer-part-3-creating-your-own-color-palettes/> [2556, November 11].
- SV Plannet Ltd (n.d.). Projector. [On-line] Available : <http://www.svplannet.com/Projector>. [2556, October 20].
- Truly Ergonomic. (n.d.). The Truly Ergonomic Mechanical Keyboard. [On-line] Available : <https://www.trulyergonomic.com/store/index.php>. [2556, October 6].
- Upstart (2012.). logo [On-line] Available : <https://www.upstart.com/>[2556, November 24].
- Weebly. (n.d.). Digitizing Tablet. [On-line] Available : <http://shaniqueeportfolio.weebly.com/digitizing-tablet.html>. [2556, October 6].
- Ziff Davis. (n.d.). Logitech Webcam c930e. [On-line] Available : <http://www.pcmag.com/article2/0,2817,2423466,00.asp>. [2556, October 6].

