



ได้ผ่านการตรวจประเมินคุณภาพหนังสือเรียนอาชีวศึกษา หลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2562  
ประจำปีงบประมาณ พ.ศ. 2566 ประกาศลำดับที่ 20

20901-1002

หนังสือเล่มนี้เรียบเรียงตามจุดประสังค์รายวิชา สmentionรายวิชา และคำอธิบายรายวิชา

หลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2562

ของสำนักงานคณะกรรมการการอาชีวศึกษา กระทรวงศึกษาธิการ

การเขียน

# ปรัชญา คอมพิวเตอร์

Basic Computer Programming

เบื้องต้น



หนังสือ 4 สี  
ก้างเล่ม

โภมล ศิริสมบูรณ์เวช

148.-

# การเขียนโปรแกรมคอมพิวเตอร์ เบื้องต้น

20901-1002

ผู้แต่ง : โภมล ศิริสมบูรณ์เวช  
พิสูจน์อักษร : อาบนนท์ ปั่นสุกาน

All Rights Reserved.  
สงวนลิขสิทธิ์ตามพระราชบัญญัติ

ปีที่พิมพ์ : 2566  
พิมพ์ครั้งที่ 1 : 3,000 เล่ม  
ISBN : 978-616-579-254-7  
ราคา : 148 บาท



สำนักพิมพ์ศูนย์ส่งเสริมอาชีวะ

89 ถนนเมห์ธรา แขวงช้างเผือก กรุงเทพฯ 10200

โทร. 0-2224-1129, 0-2224-1197

พิมพ์ที่ : บริษัท รัตน์การพิมพ์ จำกัด  
32/10 หมู่ที่ 3 ตำบลบางขุน อำเภอเกว榜กรวย จังหวัดนนทบุรี 11130  
ผู้พิมพ์ผู้ໂນຍາ 2566

# ๑๑ จุดประสงค์ สมรรถนะและคำอธิบายรายวิชา

รหัสวิชา 20901-1002 ชื่อวิชา การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น (Basic Computer Programming)

จำนวน ก-ป-น 1-2-2

หลักสูตร ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เทคโนโลยีสารสนเทศ

## จุดประสงค์รายวิชา เพื่อให้

- เข้าใจเกี่ยวกับหลักการเขียนโปรแกรม
- สามารถเขียนโปรแกรมประยุกต์ขนาดเล็ก ด้วยภาษาโปรแกรมคอมพิวเตอร์
- มีคุณธรรมจริยธรรม และค่านิยมที่ดีในการเขียนโปรแกรมคอมพิวเตอร์

## สมรรถนะรายวิชา

- แสดงความรู้เกี่ยวกับหลักการเขียนโปรแกรม
- เขียนโปรแกรมประยุกต์ขนาดเล็ก ด้วยภาษาโปรแกรมคอมพิวเตอร์

## คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการเขียนโปรแกรม การเขียนผังงาน (Flowchart) การเขียนโปรแกรมด้วยภาษาโปรแกรมคอมพิวเตอร์ การตรวจสอบและทดสอบโปรแกรม

# ตารางวิเคราะห์สมรรถนะรายวิชา

วิชา การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น  
ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

รหัสวิชา 20901-1002  
สาขา วิชาเทคโนโลยีสารสนเทศ

ตารางวิเคราะห์สมรรถนะ

หน่วยที่

สมรรถนะ  
รายวิชา

สมรรถนะรายวิชา		ผลลัพธ์ทางการศึกษาที่ยั่งยืน	มาตรฐานที่ส่งเสริมการเรียนรู้และพัฒนาศักยภาพบุคคล
1	ความรู้พื้นฐานในการเขียนโปรแกรม	✓	
2	การเขียนโปรแกรมด้วยภาษา C	✓	
3	การเขียนโปรแกรมแบบลำดับ	✓	
4	การเขียนโปรแกรมแบบเลือกทำ	✓	
5	การเขียนโปรแกรมแบบทำซ้ำ	✓	
6	ตัวแปรอาร์เรย์และสตริง	✓	
7	การเขียนและใช้งานฟังก์ชัน	✓	
8	ตัวแปรโครงสร้างและตัวแปรไฟล์	✓	
9	การพัฒนาโปรแกรมอย่างจ่าย	✓	✓



កំហា

หนังสือ “การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น” รหัสวิชา 20901-1002 นี้ได้เรียบเรียงตามหลักสูตร ประกาศนียบัตรวิชาชีพ (ปวช.) พุทธศักราช 2562 ของสาขาวิชาเทคโนโลยีสารสนเทศ โดยมีจุดประสงค์เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับหลักการเขียนโปรแกรม สามารถวิเคราะห์ ออกแบบ และเขียนขั้นตอนการแก้ไขปัญหา รวมทั้งสามารถเขียนโปรแกรมโดยใช้คำสั่งควบคุม การทำงานเบื้องต้นด้วยภาษาคอมพิวเตอร์ หนังสือเล่มนี้แบ่งหน่วยการเรียนเป็น 9 หน่วย โดยแต่ละหน่วยจะประกอบด้วยเนื้อหาความรู้ กิจกรรมการเรียนรู้ คำถามท้ายหน่วย และแบบทดสอบ เพื่อส่งเสริมให้ผู้เรียนได้ค้นคว้าแก้ปัญหาทำงานเป็นทีม และฝึกฝนปฏิบัติงานในสภาพจริงตามแนวทางของ การอาชีวศึกษา

แนวทางของหน้าสือเล่นนี้จะเน้นการเรียนรู้จากการฝึกปฏิบัติและสอดแทรกเนื้อหาบางส่วนที่ควรรู้ไว้ในส่วนของการปฏิบัติงาน เพื่อให้ผู้เรียนได้ฝึกคิดและฝึกเรียนรู้จากการฝึกปฏิบัติจริงโดยมีผู้สอนเป็นผู้คุยให้คำแนะนำ (facilitator)

ผู้เรียนเริ่มขอขอบคุณสำนักพิมพ์ที่ให้โอกาสในการจัดพิมพ์หนังสือเล่มนี้เพื่อแบ่งปันและแลกเปลี่ยนความรู้ในวงกว้าง และขอขอบคุณ คุณครุภิวิสรา อับดุลลาติฟ ที่ได้ให้แนวคิดการจัดกิจกรรมการเรียนการสอนโปรแกรมแบบ BokplusB เพื่อสร้างหลักการพื้นฐานในการเขียนโปรแกรมคอมพิวเตอร์ ผ่านกิจกรรมการเรียนรู้โดยไม่ใช้คอมพิวเตอร์

ผู้เรียบเรียงหวังเป็นอย่างยิ่งว่าหนังสือพื้นฐานการเขียนโปรแกรมเล่มนี้จะเป็นประโยชน์ในการจัดการเรียนการสอนและเกิดประโยชน์ต่อผู้สนใจทั่วไป หากมีข้อผิดพลาดหรือข้อเสนอแนะ ประการใดผู้เรียบเรียงขอน้อมรับเพื่อปรับปรุงเนื้อหาให้สมบูรณ์ยิ่งขึ้นต่อไป สำหรับความดีที่จะพึงมี ขอขอบคุณบุพการีที่สนับสนุนด้านการศึกษา และครูอาจารย์ที่ให้ความรู้ทางวิชาการ

## ໂຄມລ ຕີຣີສມບູຮນໍເວຊ

<b>หน่วยที่ 1 ความรู้พื้นฐานในการเขียนโปรแกรม</b>	<b>1</b>
ความสำคัญและหมายของโปรแกรมคอมพิวเตอร์	3
ภาษาคอมพิวเตอร์	3
ตัวแปลภาษา	5
แบบคิดในการเขียนโปรแกรม	6
รูปแบบการควบคุมการทำงานของโปรแกรม	6
ขั้นตอนการพัฒนาโปรแกรม	8
รหัสเกี่ยม และผังงานโปรแกรม	11
กิจกรรมการเรียนรู้ หน่วยที่ 1	16
แบบฝึกหัดหน่วยที่ 1	26
แบบทดสอบ หน่วยที่ 1	27
<b>หน่วยที่ 2 การเขียนโปรแกรมด้วยภาษา C</b>	<b>29</b>
โครงสร้างภาษาคอมพิวเตอร์	31
แบบนำภาษา C	32
โครงสร้างของโปรแกรมภาษา C	33
องค์ประกอบของโปรแกรม	34
คำสั่งรับข้อมูลและแสดงผลในภาษา C	39
เครื่องมือในการพัฒนาโปรแกรมภาษา C	43
กิจกรรมการเรียนรู้ หน่วยที่ 2	45
แบบฝึกหัดหน่วยที่ 2	49
แบบทดสอบ หน่วยที่ 2	51
<b>หน่วยที่ 3 การเขียนโปรแกรมแบบตามลำดับ</b>	<b>53</b>
โปรแกรมแบบลำดับ	55
การคำนวนค่าในภาษา C	55
ลำดับการคำนวน	57
การแปลงแบบข้อมูล	58
การปัดเศษและแปลงเลขฐาน	60
กิจกรรมการเรียนรู้ หน่วยที่ 3	62
แบบฝึกหัดหน่วยที่ 3	66
แบบทดสอบ หน่วยที่ 3	70

<b>หน่วยที่ 4 การเขียนโปรแกรมแบบเลือกทำ</b>	<b>73</b>
การเลือกทำในรูปแบบ if	74
การเลือกทำในรูปแบบ if...else	80
การเลือกทำโดยใช้รูปแบบ switch	82
โปรแกรมในลักษณะเงื่อนไขซ้อน	86
กิจกรรมการเรียนรู้ หน่วยที่ 4	90
แบบฝึกหัดหน่วยที่ 4	99
แบบทดสอบ หน่วยที่ 4	103
<b>หน่วยที่ 5 การเขียนโปรแกรมแบบกำช้ำ</b>	<b>107</b>
การกำช้ำ แบบ for	108
การกำช้ำ แบบ while	112
การกำช้ำ แบบ do...while	114
การเขียนลำดับการทำงานแบบกำช้ำ	118
กิจกรรมการเรียนรู้ หน่วยที่ 5	121
แบบฝึกหัดหน่วยที่ 5	128
แบบทดสอบ หน่วยที่ 5	131
<b>หน่วยที่ 6 ตัวแปรอาร์เรย์และสตริง</b>	<b>135</b>
ตัวแปรอาร์เรย์	136
ตัวแปรอาร์เรย์หลายมิติ	140
การกำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์	144
ตัวแปรสตริง	146
ฟังก์ชันเกี่ยวกับสตริง (String Function)	150
กิจกรรมการเรียนรู้ หน่วยที่ 6	155
แบบฝึกหัดหน่วยที่ 6	160
แบบทดสอบ หน่วยที่ 6	163
<b>หน่วยที่ 7 การเขียนและใช้งานฟังก์ชัน</b>	<b>167</b>
จุดประสงค์ของการเขียนฟังก์ชัน	169
การเขียนฟังก์ชันแบบไม่ผ่านค่า	170
การเขียนฟังก์ชันผ่านค่าทางเดียว	172
การเขียนฟังก์ชันผ่านค่าประกอบ	174
การส่งอาร์เรย์ให้กับฟังก์ชัน	177
ตัวแบบฟังก์ชัน	179
ตัวแปรร่วมและตัวแปรประจำที่	180

กิจกรรมการเรียนรู้ หน่วยที่ 7	182
แบบฝึกหัดหน่วยที่ 7	190
แบบทดสอบ หน่วยที่ 7	192
<b>หน่วยที่ 8 ตัวแปรโครงสร้างและตัวแปรไฟฟ้า</b>	<b>195</b>
การใช้งานตัวแปรโครงสร้าง	196
การใช้งานตัวแปรไฟฟ้า	200
กิจกรรมการเรียนรู้ หน่วยที่ 8	207
แบบฝึกหัดหน่วยที่ 8	211
แบบทดสอบ หน่วยที่ 8	212
<b>หน่วยที่ 9 การพัฒนาโปรแกรมอย่างง่าย</b>	<b>215</b>
แนวทางในการพัฒนาโปรแกรม	216
ข้อผิดพลาดในการพัฒนาโปรแกรม (Error)	217
ตัวอย่างการพัฒนาโปรแกรมอย่างง่าย	219
กิจกรรมการเรียนรู้ หน่วยที่ 9	232
แบบฝึกหัดหน่วยที่ 9	233
แบบทดสอบ หน่วยที่ 9	234
<b>ภาคผนวก</b>	<b>237</b>
<b>บรรณานุกรม</b>	<b>245</b>

# ความรู้พื้นฐาน ในการเขียนโปรแกรม



## สาระสำคัญ

การเขียนโปรแกรมคอมพิวเตอร์ คือ การเรียนรู้และใช้ภาษาคอมพิวเตอร์ในการสื่อสารกับเครื่องคอมพิวเตอร์ ทำงานตามวัตถุประสงค์ที่ต้องการโดยใช้ภาษาคอมพิวเตอร์ในการเขียนโปรแกรม การเขียนโปรแกรม ที่ดีจะต้องทำอย่างมีหลักการและเป็นระบบตามขั้นตอนการพัฒนาโปรแกรม เพื่อให้โปรแกรมมีความถูกต้องและสะดวกต่อการปรับปรุงในภายหลัง

## สารการเรียนรู้

- ความสำคัญและความหมายของโปรแกรมคอมพิวเตอร์
- ภาษาคอมพิวเตอร์
- ตัวแปลงภาษา
- แนวคิดในการเขียนโปรแกรม
- ขั้นตอนการพัฒนาโปรแกรม
- รหัสเทียมและผังงานโปรแกรม

## จุดประสงค์เชิงพฤติกรรม

- บอกรความหมายและความสำคัญของโปรแกรมคอมพิวเตอร์ได้
- บอกรความหมายของภาษาคอมพิวเตอร์
- บอกรความแตกต่างการแปลงภาษาของตัวแปลงภาษาแต่ละแบบได้
- บอกรักษณะการเขียนโปรแกรมตามแนวคิดทั้ง 2 แบบได้
- อธิบายขั้นตอนในการพัฒนาโปรแกรมได้
- ใช้รหัสเทียมและผังงานโปรแกรมในการแก้ปัญหาได้

## สมรรถนะอาชีพ

แสดงความรู้เกี่ยวกับโปรแกรมคอมพิวเตอร์



### 2 ความรู้พื้นฐานในการเขียนโปรแกรม



## ความสำคัญและความหมายของโปรแกรมคอมพิวเตอร์

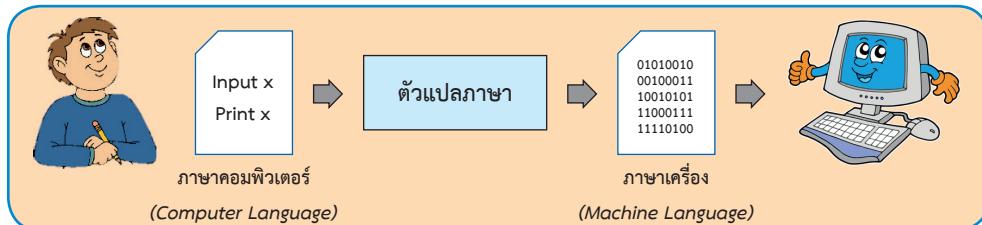
คอมพิวเตอร์จัดว่าเป็นอุปกรณ์อิเล็กทรอนิกส์ที่แตกต่างจากเครื่องใช้อิเล็กทรอนิกส์ทั่วไป เพราะเครื่องใช้อิเล็กทรอนิกส์ส่วนใหญ่จะทำงานตามการออกแบบของร่างกายที่กำหนดไว้ การปรับเปลี่ยนการทำงานจำเป็นต้องออกแบบและเปลี่ยนอุปกรณ์ในวงจรใหม่ทั้งหมด ในขณะที่คอมพิวเตอร์จะทำงานตามชุดคำสั่งหรือโปรแกรมที่กำหนดไว้ การปรับเปลี่ยนการทำงานจึงทำได้ง่ายโดยไม่จำเป็นต้องเปลี่ยนอุปกรณ์ทางด้านฮาร์ดแวร์ ดังเช่นเครื่องคอมพิวเตอร์ที่ใช้งานทั่วไปอาจใช้เป็นเครื่องที่ทำงานแทนเครื่องพิมพ์ดีด หรือเป็นเครื่องเล่นเกมต่าง ๆ ขึ้นอยู่กับการติดตั้งและเรียกใช้โปรแกรมตามความต้องการ

สำหรับความหมายของ “**โปรแกรมคอมพิวเตอร์**” จะหมายถึงชุดคำสั่งที่ถูกจัดวางเพื่อส่งงานคอมพิวเตอร์ให้ทำงานตามเงื่อนไขหรือตามจุดประสงค์ที่กำหนดไว้ เช่นเดียวกับโปรแกรมภาษาพยนตร์ตามโครงสร้างที่จัดฉายภาษาพยนตร์ตามลำดับที่กำหนดไว้ เป็นต้น ดังนั้นจึงอาจสรุปได้ว่า **การเขียนโปรแกรม** คือ การเรียบเรียงหรือจัดเรียงคำสั่งเพื่อส่งงานคอมพิวเตอร์ให้ทำงานตามจุดประสงค์ที่ต้องการนั้นเอง



## ภาษาคอมพิวเตอร์

โดยทั่วไปคอมพิวเตอร์จะรับรู้การสั่งงานโดยใช้ข้อมูลที่เป็นสัญญาณดิจิตอลถ่ายกับเลขฐานสอง (Binary) ประกอบด้วยตัวเลข 0 และ 1 ซึ่งเราเรียกชุดข้อมูลดังกล่าวว่า **รหัสภาษาเครื่อง (Machine Code หรือ Machine Language)** การสั่งงานโดยใช้ภาษาเครื่องเป็นเรื่องยากต่อการอ่านและเขียนโปรแกรม ดังนั้นจึงมีการกำหนดคำสั่งที่เป็นตัวอักษรมาใช้แทนตัวเลขเพื่อให้เขียนเป็นโปรแกรมได้ง่ายขึ้นเรียกว่า **ภาษาคอมพิวเตอร์ (Computer Language)** โดยเมื่อเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์เสร็จเรียบร้อยแล้ว ก็จะนำโปรแกรมที่เขียนขึ้นมาแปลเป็น ภาษาเครื่อง โดยใช้ตัวแปลภาษาจากนั้นจึงนำไปสั่งงานคอมพิวเตอร์อีกต่อหนึ่ง ดังรูปที่ 1.1



**รูปที่ 1.1** แสดงการสั่งงานคอมพิวเตอร์โดยใช้ภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมมีด้วยกันหลายภาษาแต่อาจแบ่งได้เป็น 2 ระดับ คือ

## 1) ภาษาระดับต่ำ (Low level Computer Language)

ภาษาระดับต่ำ คือภาษาที่แปลงจากภาษาเครื่องมาเป็นภาษาคอมพิวเตอร์แบบง่าย ๆ ตรงไปตรงมา เช่น **ภาษาแอสเซมบลี (Assembly)** สำหรับข้อเสียของโปรแกรมที่เขียนด้วยภาษาระดับต่ำ คือจะผูกติดกับชิปปี้และสาร์ดแวร์ของคอมพิวเตอร์เครื่องนั้น ถ้ามีการเปลี่ยนชาร์ดแวร์ของเครื่องคอมพิวเตอร์ หรือถ้านำไปใช้กับชิปปี้คนละตรากฎอาจใช้งานไม่ได้ ทำให้ต้องเขียนโปรแกรมขึ้นมาใหม่ ทั้งหมด สำหรับข้อดี คือสามารถแปลงชุดคำสั่งเป็นภาษาเครื่องได้อย่างกระหึ่รัด จึงทำให้ประมวลผลได้รวดเร็วและใช้หน่วยความจำน้อย อย่างไรก็ได้การเขียนโปรแกรมด้วยภาษาระดับต่ำค่อนข้างมีความยุ่งยากโดยเฉพาะโปรแกรมขนาดใหญ่ จึงนิยมนำไปใช้กับการเขียนโปรแกรมควบคุมอุปกรณ์ในงานอิเล็กทรอนิกส์หรืองานอุตสาหกรรมมากกว่างานทางธุรกิจ แต่ปัจจุบันเนื่องจากชิปประมวลผลหรือชิปปี้มีความเร็วสูงขึ้นมากอีกทั้งหน่วยความจำมีราคาถูกลง ทำให้ข้อจำกัดดังกล่าวหมดไป จนสามารถใช้ภาษาระดับสูงในการเขียนโปรแกรมสำหรับงานควบคุมได้โดยให้ผลไม่แตกต่างกัน ปัจจุบันจึงนิยมใช้ภาษาระดับสูงมาแทนการใช้ภาษาระดับต่ำหรือแอสเซมบลี

## 2) ภาษาระดับสูง (High level Computer Language)

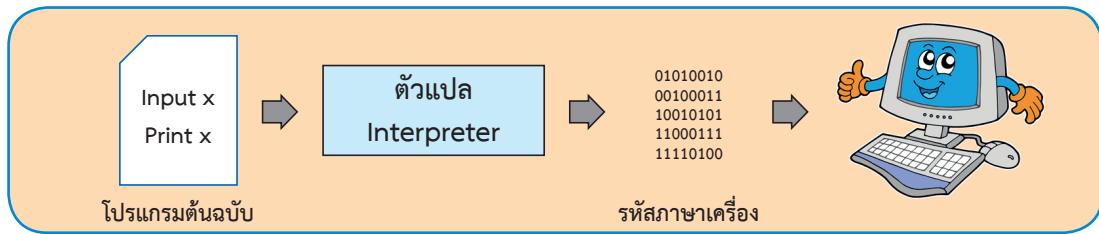
ภาษาระดับสูง เป็นภาษาที่ไม่ผูกติดกับชาร์ดแวร์ สามารถนำไปใช้ในเครื่องคอมพิวเตอร์ที่ใช้ชิปปี้ต่างกันได้อย่างไม่มีปัญหา และมีความสะดวกต่อการเขียนโปรแกรมอย่างมากจึงนิยมนำมาใช้ในการเขียนโปรแกรม สำหรับภาษาระดับสูงมีให้เลือกใช้งานหลายภาษา เช่นในอดีตซึ่งเขียนโปรแกรมในลักษณะตัวอักษรหรือเทกซ์mode (Text Mode) และใช้การติดต่อผ่านแป้นพิมพ์ ภาษาที่นิยมใช้ได้แก่ ภาษา BASIC ภาษา Pascal ภาษา Fortran, ภาษา Cobol และภาษา C จากนี้ได้พัฒนามาเป็นการเขียนโปรแกรมติดต่อผู้ใช้ในลักษณะกราฟิกใหม่ เช่นที่ใช้งานในระบบปฏิบัติการวินโดวส์ เรียกว่า GUI (Graphic User Interface) โดยใช้วิธีคลิกเมาส์บนหน้าจอภาพ และทำงานตามเหตุการณ์ที่กำหนดไว้ (Event Driven) ภาษาที่นำมาใช้ในการเขียนโปรแกรมในลักษณะดังกล่าว ได้แก่ Visual BASIC และ Visual C สำหรับปัจจุบันซึ่งนิยมใช้งานโปรแกรมผ่านระบบเว็บ (Web Application) และโปรแกรมที่ใช้งานบนโทรศัพท์ (Mobile Application) ซึ่งโปรแกรมดังกล่าวจะด้วยเขียนด้วยภาษาสคริปต์ (Script) เช่น PHP และ JavaScript ร่วมกับการใช้ส่วนขยายที่เรียกว่าเฟรมเวิร์ก (Framework) แต่ไม่ว่าจะเขียนโปรแกรมด้วยภาษาใดก็ตาม จะเป็นต้องมีความรู้และเข้าใจหลักการเบื้องต้นในการเขียนโปรแกรม



## ตัวแปลภาษา

การนำโปรแกรมที่เขียนด้วยภาษาคอมพิวเตอร์ไปใช้งาน จะต้องนำมาแปลให้เป็นคำสั่งภาษาเครื่อง (Machine Language) เสียก่อนด้วยตัวแปลภาษา ซึ่งสามารถแบ่งตามลักษณะการแปลได้ 2 แบบ คือ

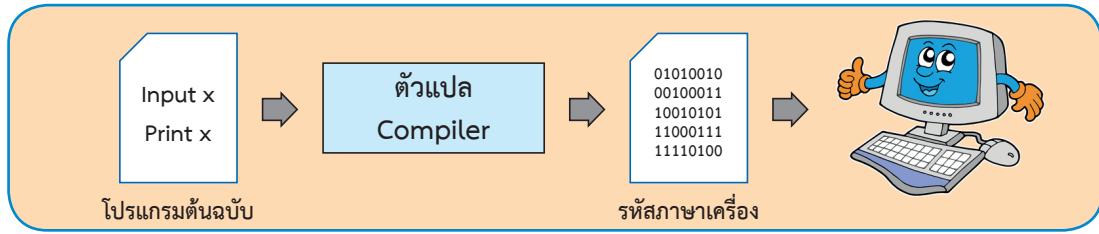
**1) แบบอินเทอร์เพรเตอร์ (Interpreter)** มีลักษณะแปลทีละคำสั่งแล้วสั่งงาน โดยอ่านคำสั่งในภาษาคอมพิวเตอร์ทีละคำสั่งมาแปลเป็นภาษาเครื่องแล้วให้คอมพิวเตอร์ประมวลผลตามคำสั่ง ดังกล่าว จนกว่าจะอ่านคำสั่งถัดไปมาแปลและสั่งงานในลักษณะเดียวกันจนจบโปรแกรม แต่เมื่อพบข้อผิดพลาดระหว่างการแปลโปรแกรมจะหยุดทำงานทันที ดังรูปที่ 1.2



รูปที่ 1.2 การแปลแบบอินเทอร์เพรเตอร์ (Interpreter)

ข้อเสียของตัวแปลภาษาแบบอินเทอร์เพรเตอร์ คือ ทุกครั้งที่ใช้งานจะต้องนำโปรแกรมต้นฉบับมาแปลใหม่ จึงทำงานได้ช้า แต่ข้อดีคือสามารถตรวจสอบแก้ไขโปรแกรมได้ง่าย

**2) แบบคอมไพล์เตอร์ (Compiler)** เป็นการแปลคำสั่งทั้งหมดของโปรแกรมในคราวเดียวให้เป็นไฟล์ประมวลผล (Execute File) และจึงนำไปใช้ได้โดยไม่ต้องมาแปลใหม่หรืออาศัยโปรแกรมต้นฉบับอีก ดังรูปที่ 1.3



รูปที่ 1.3 การแปลแบบคอมไпал์เตอร์ (Compiler)

จะเห็นว่าข้อดีของตัวแปลภาษาแบบคอมไпал์เตอร์ คือ เมื่อต้องการสั่งงานอีกครั้งก็ไม่ต้องนำโปรแกรมต้นฉบับมาแปลซ้ำเหมือนตัวแปลแบบอินเทอร์เพรเตอร์ จึงทำงานได้รวดเร็ว แต่การตรวจสอบและแก้ไขโปรแกรมอาจทำได้ยากกว่า



## แนวคิดในการเขียนโปรแกรม

สำหรับแนวคิดในการเขียนโปรแกรมอาจแบ่งเป็น 2 ลักษณะดังนี้

### 1) การเขียนโปรแกรมแบบเชิงกระบวนการ (Procedural Programming)

เป็นแนวคิดในการแบ่งโปรแกรมเป็นส่วนย่อยตามหน้าที่การทำงาน จักนั้นจึงนำโปรแกรมส่วนย่อยที่เขียนขึ้นมารวมกันเป็นโปรแกรมทั้งระบบโดยเรียกผ่านโปรแกรมหลัก การเขียนโปรแกรมแบบนี้จะแบ่งงานออกเป็นส่วนของโปรแกรมย่อย (Procedure) และฟังก์ชัน (Function) ทำให้ตรวจสอบและแก้ไขได้ง่าย อย่างไรก็ถือเป็นระบบงานขนาดใหญ่จะเกิดปัญหาในการขยายความสามารถและการตรวจสอบแก้ไขโปรแกรม ดังนั้นการเขียนโปรแกรมลักษณะนี้จึงเหมาะสมสำหรับงานโปรแกรมขนาดเล็ก หรือสำหรับการเรียนรู้การเขียนโปรแกรมในระดับต้น

### 2) การเขียนโปรแกรมแบบเชิงวัตถุ (OOP : Object Oriented Programming)

การเขียนโปรแกรมแบบเชิงวัตถุ เป็นแนวคิดแบบใหม่ในการพัฒนาโปรแกรม โดยจะมองส่วนต่าง ๆ ของโปรแกรมเป็นวัตถุซึ่งมีทั้งคุณสมบัติ (Properties) และความสามารถในการกระทำ (Method) โดยมีข้อดีคือสามารถนำโปรแกรมที่เขียนไปต่อ�อดใช้งานในภายหลังได้สะดวกกว่า การเขียนโปรแกรมแบบเชิงกระบวนการ แต่การออกแบบโปรแกรมจะมีความซับซ้อนและต้องใช้ความละเอียดในการวางแผนมากกว่า

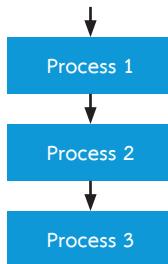


## รูปแบบการควบคุมการทำงานของโปรแกรม

การเขียนโปรแกรมส่วนใหญ่นิยมเขียนเป็นโครงสร้าง (Structure Programming) เพื่อให้อ่านโปรแกรมได้ง่าย และสะดวกต่อการแก้ไข โดยมีรูปแบบสำหรับควบคุมการทำงานของโปรแกรม 3 รูปแบบ ได้แก่

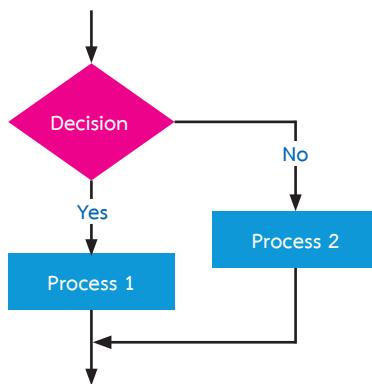
### 1) รูปแบบทำงานตามลำดับ (Sequence)

เป็นการกำหนดการทำงานของโปรแกรมให้ทำงานเรียงตามลำดับคำสั่งที่วางไว้ ดังรูปที่ 1.4



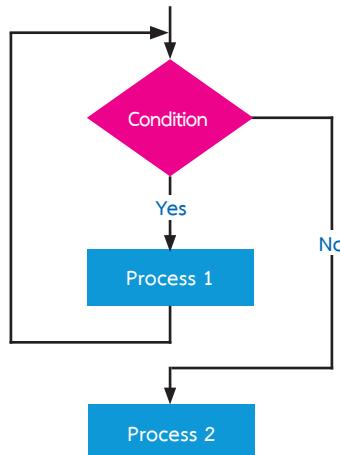
รูปที่ 1.4 แสดงรูปแบบการทำงานแบบตามลำดับ (Sequence)

**2) รูปแบบเลือกทำ (Selection) หรือตัดสินใจ (Decision)** เป็นการกำหนดให้โปรแกรมเลือกทำงานตามเงื่อนไขที่นำมาตรวจสอบ ดังรูปที่ 1.5



รูปที่ 1.5 แสดงรูปแบบการทำงานแบบเลือกทำ (Selection)

**3) รูปแบบทำซ้ำ (Repetition)** คือ การควบคุมให้โปรแกรมวนกลับมาทำงานซ้ำๆ ตามเงื่อนไขที่กำหนดไว้ ดังรูปที่ 1.6



รูปที่ 1.6 แสดงรูปแบบการทำงานแบบทำซ้ำ (Repetition)

สำหรับรายละเอียดการควบคุมแต่ละแบบจะได้แบ่งกล่าวในหน่วยการเรียนถัดไป



## ขั้นตอนการพัฒนาโปรแกรม

การเขียนโปรแกรมคอมพิวเตอร์จัดเป็นงานที่ต้องวางแผนและทำตามขั้นตอน โดยส่วนใหญ่ต้องคิดวิเคราะห์และวางแผนแนวทางในการเขียนโปรแกรมเสียก่อนเพื่อให้โปรแกรมตอบสนองต่อความต้องการของผู้ใช้งาน และมีความเหมาะสมกับระบบงานที่ทำ การสร้างหรือพัฒนาโปรแกรมคอมพิวเตอร์ในลักษณะทำไปแก้ไปอาจทำได้แต่อาจทำให้เกิดปัญหาในภายหลังจนต้องกลับมาเขียนและออกแบบใหม่ทั้งหมด ดังนั้นการพัฒนาโปรแกรมที่เหมาะสมจะมีการพัฒนาตามขั้นตอนมาตรฐานซึ่งจะช่วยลดปัญหาความผิดพลาดที่จะเกิดขึ้น และช่วยให้การแก้ไขหรือปรับปรุงโปรแกรมภายหลังทำได้สะดวกขึ้น สำหรับขั้นตอนการพัฒนาโปรแกรมในขั้นตอนอาจแบ่งเป็น 5 ขั้นตอนดังนี้

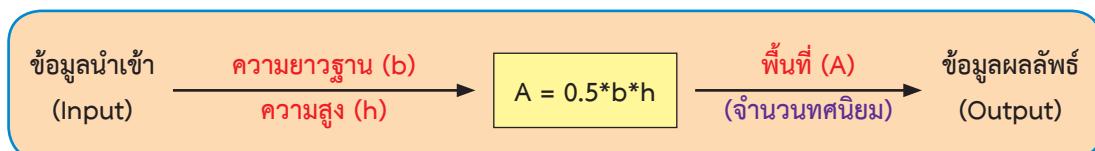
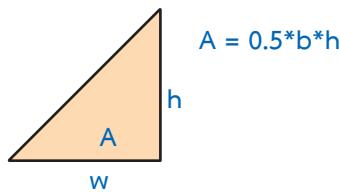
- 1) วิเคราะห์ปัญหา (Problem Analysis)
- 2) ออกแบบโปรแกรม (Design)
- 3) เขียนโปรแกรม (Coding)
- 4) ทดสอบโปรแกรม (Testing)
- 5) รวบรวมเอกสาร (Documentation)
- 6) บำรุงรักษา (Maintenance)

### 1) การวิเคราะห์ปัญหา (Problem Analysis)

การวิเคราะห์ปัญหาเป็นขั้นตอนที่มีความสำคัญในการพัฒนาโปรแกรม ถ้าวิเคราะห์ปัญหาไม่ถูกต้องอาจทำให้โปรแกรมที่เขียนขึ้นไม่ครอบคลุมความต้องการของผู้ใช้งานอาจต้องนำกลับมาแก้ไขหรือเขียนใหม่ทั้งหมด โดยทั่วไปการวิเคราะห์ปัญหาจะเริ่มจากการสอบถามข้อมูลจากผู้ใช้งาน และจากรายงานต่าง ๆ ที่ผู้ใช้งานต้องการ เพื่อนำมาพิจารณาข้อมูลนำเข้า (Input) ที่ต้องใช้ และข้อมูลผลลัพธ์ (Output) ที่ต้องการ จากนั้นจึงนำมากำหนดเป็นตัวแปรเก็บข้อมูลในโปรแกรม รวมไปถึงทำความสัมพันธ์หรือวิธีการคำนวนค่าผลลัพธ์จากข้อมูลนำเข้าที่กำหนด สำหรับการศึกษาการเขียนโปรแกรมในระดับต้น ส่วนใหญ่จะเป็นการแก้โจทย์ปัญหาของโปรแกรม จึงมีขั้นตอนในวิเคราะห์และแก้ปัญหาดังนี้

- พิจารณาข้อมูลผลลัพธ์ที่ต้องการหรือเอาต์พุตของโปรแกรมคืออะไร
- วิเคราะห์ว่าต้องใช้ข้อมูลนำเข้า (input) อะไรที่จะนำมาหาค่าผลลัพธ์ (output)
- กำหนดตัวแปร และค่าคงที่ที่จะใช้ในโปรแกรม
- กำหนดวิธีคำนวนค่าเพื่อหาผลลัพธ์

**ตัวอย่าง** การวิเคราะห์งานโปรแกรมคำนวณพื้นที่สามเหลี่ยม สามารถวิเคราะห์ได้ดังนี้



รูปที่ 1.7 แสดงการความล้มเหลวในการคำนวณพื้นที่สามเหลี่ยม

- 1) ผลลัพธ์ที่ต้องการ หรือเอาต์พุตของโปรแกรม คือ พื้นที่ของสามเหลี่ยม
- 2) ข้อมูลนำเข้าที่ต้องใช้เพื่อหาผลลัพธ์ คือ ความยาวฐาน และ ความสูง ของสามเหลี่ยม โดย พื้นที่สามเหลี่ยม =  $0.5 \times \text{ความยาวฐาน} \times \text{ความสูง}$
- 3) กำหนดตัวแปร และค่าคงที่ที่ใช้ในโปรแกรม
  - 3.1 ตัวแปรอินพุต คือ กำหนดให้ ตัวแปร **b** เก็บค่า ความยาวฐาน กำหนดเป็น เลขจำนวนเต็ม
  - 3.2 ตัวแปร **h** เก็บค่า ความสูง กำหนดเป็น เลขจำนวนเต็ม
- 4) วิธีคำนวณค่าเพื่อหาผลลัพธ์ คือ  $A = 0.5 * b * h$

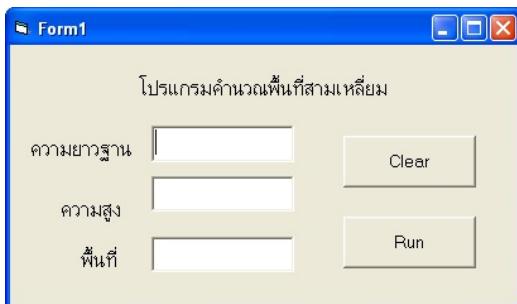
## 2) การออกแบบโปรแกรม (Program Design)

ขั้นตอนการออกแบบโปรแกรม เป็นการพิจารณาและวางแผนแนวทาง ทั้งในส่วนของการออกแบบหน้าจอแสดงผล และการกำหนดขั้นตอนการทำงานของโปรแกรม (Algorithm) เช่น โปรแกรมคำนวณพื้นที่รูปสามเหลี่ยม ในกรณีเป็นโปรแกรมแบบเก่าหรือแบบเท็กซ์mode (Text Mode) ซึ่งใช้การติดต่อด้วยแป้นพิมพ์เพียงอย่างเดียว การออกแบบหน้าจอแสดงผล สามารถออกแบบในลักษณะดังรูปที่ 1.8

Area of Triangle	//แสดงชื่อโปรแกรม
Base = xx	//รับข้อมูลความยาวฐานเก็บในตัวแปร b
Height = xx	//รับข้อมูลความสูงเก็บในตัวแปร h
Area = xxx.xx	//แสดงผลลัพธ์ในตัวแปร A

 รูปที่ 1.8 แสดงการออกแบบหน้าจอโปรแกรมแบบเท็กซ์โหมด

สำหรับการใช้งานที่ใช้การติดต่อโดยใช้มาสคิลิกบนหน้าจอ ซึ่งเป็นการติดต่อผู้ใช้แบบกราฟิกโหมด หรือ GUI (Graphic User Interface) อาจกำหนดการแสดงผลในลักษณะดังรูปที่ 1.9



 รูปที่ 1.9 แสดงการออกแบบหน้าจอโปรแกรมแบบกราฟิกโหมด

จากนั้นจะเขียนขั้นตอนการทำงานของโปรแกรม โดยเขียนเป็นประโยคคำสั่งทีละบรรทัดเรียกว่า รหัสเทียม (Pseudo Code) หรืออาจเขียนเป็น ผังงาน (Flowchart) โดยใช้สัญลักษณ์ประกอบ

### 3) การเขียนโปรแกรม (Coding)

เป็นขั้นตอนในการนำรหัสเทียมหรือผังงานโปรแกรม มาเขียนเป็นรหัสโปรแกรมด้วยชุดคำสั่งภาษาคอมพิวเตอร์ตามรูปแบบและโครงสร้างของภาษาหนึ่ง ๆ เช่นกรณีเขียนโปรแกรมโดยใช้ภาษา C จะมีรูปแบบโปรแกรมและคำสั่ง ดังรูปที่ 1.10

```
#include <stdio.h>
main()
{
    int b,h;
    float A;
    printf("Area of Triangle \n");
    printf("Base = ");scanf("%d",&b);
    printf("Height = ");scanf("%d",&h);
    A = 0.5*b*h;
    printf("Area = %3.2f \n",A);
}
```

 รูปที่ 1.10 แสดงโปรแกรมคำนวณพื้นที่สามเหลี่ยมที่เขียนด้วยภาษา C

#### 4) การทดสอบโปรแกรม (Testing)

หลังจากเขียนโปรแกรมเสร็จสิ้น ก็จะนำโปรแกรมมาทดสอบเพื่อตรวจสอบความถูกต้องของโปรแกรม โดยทดลองป้อนค่าข้อมูลตามที่โปรแกรมกำหนด จากนั้นนำผลลัพธ์ที่ได้มาเปรียบเทียบกับการคำนวณค่าด้วยมือ หรือผลลัพธ์จากรายงานที่เคยทำ ว่าถูกต้องตรงกันหรือไม่ ถ้ายังมีข้อบกพร่อง ก็จะนำมาแก้ไขแล้วจึงนำไปให้ผู้ใช้งานทดลองใช้

#### 5) การรวมรวมเอกสาร (Documentation)

โดยทั่วไปหลังจากจัดทำโปรแกรมเสร็จสิ้น ก็จะจัดเก็บและรวบรวมเอกสารการดำเนินการต่าง ๆ เช่น เอกสารการวิเคราะห์งาน รหัสเทียมหรือผังงานโปรแกรม โปรแกรมต้นฉบับ (Source Code) รวมทั้งปัญหาที่พบระหว่างการจัดทำโปรแกรม เพื่อนำไปใช้ในการพัฒนาแก้ไขและปรับปรุงโปรแกรมต่อไป

#### 6) การบำรุงรักษา (Maintenance)

หลังจากตรวจสอบโปรแกรมที่เขียนขึ้นผ่านการทดสอบการใช้งานจนมั่นใจ ก็จะนำโปรแกรมไปติดตั้งใช้งานในระบบจริง จากนั้นผู้เขียนโปรแกรมหรือทีมงานพัฒนาโปรแกรมจะอยู่ติดตามดูแลผู้ใช้งาน เช่น จัดฝึกอบรมการใช้งาน หรือจัดทำคู่มือการใช้งานโปรแกรมให้แก่ผู้ใช้งาน เพื่อให้ผู้ใช้สามารถใช้โปรแกรมได้อย่างถูกต้อง รวมทั้งให้ข้อมูลช่องทางการติดต่อเพื่อช่วยเหลือให้คำแนะนำแก่ผู้ใช้งาน



### รหัสเทียม และผังงานโปรแกรม

**รหัสเทียม (Pseudo Code)** และ **ผังงานโปรแกรม (Flowchart)** มักนิยมใช้เป็นเครื่องมือในการเขียนขั้นตอนการทำงานของโปรแกรม หรือที่เรียกว่า อัลกอริทึม (Algorithm) เพื่อใช้เป็นต้นแบบหรือแผนที่นำทางในการเขียนโปรแกรม สำหรับความแตกต่างระหว่างรหัสเทียมกับผังงานโปรแกรมคือ

**1) รหัสเทียม (Pseudo Code)** เป็นการเขียนการทำงานของโปรแกรมในลักษณะประโยคคำสั่งด้วยภาษาที่เข้าใจได้ง่าย โดยไม่ได้เจาะจงใช้คำสั่งหรือรูปแบบไวยากรณ์ตามภาษาคอมพิวเตอร์ แต่จะเขียนโดยใช้ภาษาทั่วไปซึ่งส่วนใหญ่นิยมเขียนเป็นภาษาอังกฤษเพื่อความเป็นสากล

## ตัวอย่างที่ 1 การเขียนลำดับการทำงานของโปรแกรมคำนวณพื้นที่สามเหลี่ยม โดยใช้รหัสเทียม

ขั้นตอน	รหัสเทียม
<b>เริ่มต้น</b> <ol style="list-style-type: none"> <li>1. ประกาศตัวแปร b และ h เป็นจำนวนเต็ม</li> <li>2. ประกาศตัวแปร Area เป็นแบบศนนิยม</li> <li>3. แสดงชื่อโปรแกรม</li> <li>4. รับข้อมูลความยาวฐาน และความสูง</li> <li>5. คำนวณค่าพื้นที่</li> <li>6. แสดงผลลัพธ์ในตัวแปรพื้นที่</li> </ol>	<b>Begin</b> Declare b, h as integer Declare Area as float print "Triangle Area" input b, h $Area = 0.5 * b * h$ print Area
<b>จบโปรแกรม</b>	<b>End</b>

## ตัวอย่างที่ 2 การเขียนลำดับการทำงานของโปรแกรมตรวจสอบการเรียน ด้วยรหัสเทียม

ขั้นตอน	รหัสเทียม
<b>เริ่มต้น</b> <ol style="list-style-type: none"> <li>1. ประกาศตัวแปร score เป็นจำนวนเต็ม</li> <li>2. รับข้อมูล คะแนนสอบ เก็บในตัวแปร score</li> <li>3. ตรวจสอบตัวแปร score น้อยกว่า 50 หรือไม่ ถ้าใช่ ให้แสดงว่า “สอบตก” ถ้าไม่ใช่ ให้แสดงว่า “สอบผ่าน”</li> </ol>	<b>Begin</b> Declare score as integer input score if score < 50 print "Fail" else print "Pass"
<b>จบโปรแกรม</b>	<b>End</b>

2) พังงาน (Flowchart) เป็นการเขียนลำดับการทำงานโดยใช้สัญลักษณ์ต่าง ๆ ประกอบกับคำอธิบายสั้น ๆ และใช้ลูกศรเป็นตัวแสดงลำดับการทำงานของโปรแกรม โดยสัญลักษณ์ที่ใช้ในพังงานโปรแกรมโดยส่วนใหญ่จะเป็นดังนี้

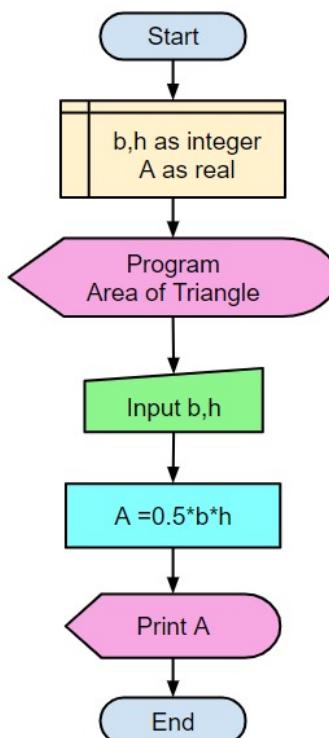
สัญลักษณ์	จุดประสงค์การใช้งาน
	จุดเริ่มต้นหรือจุดสิ้นสุด (Start /Stop Terminal)
	ใช้ประมวลตัวแปรและค่าคงที่ (Internal Storage)
	รับข้อมูลเข้าทางแป้นพิมพ์ (Manual Input)
	แสดงผลลัพธ์ทางจอภาพ (Display)
	รับข้อมูลเข้าหรือส่งข้อมูลออก (Input/Output)
	กำหนดหรือคำนวณค่า (Define/Process)
	การตัดสินใจ (Decision)
	เตรียมการทำงานแบบวนรอบ (Loop Preparation)
	เรียกใช้งานโปรแกรมย่อย (Subroutine)

สำหรับวิธีเขียนผังงานโปรแกรม มีข้อปฏิบัติดังนี้

- 1) ต้องใช้สัญลักษณ์ให้ถูกต้องตรงกับจุดประสงค์ที่ต้องการสื่อความหมาย
- 2) ข้อความในสัญลักษณ์ควรเขียนให้สั้นกระชับและเข้าใจง่าย
- 3) ต้องมีลูกศรกำกับเพื่อบอกทิศทางการทำงานของโปรแกรมอย่างชัดเจน
- 4) ลูกศรที่ใช้บอกทิศทาง ต้องเขียนจากบนลงล่างและจากซ้ายไปขวา
- 5) ถ้าต้องลากเส้นเชื่อมโดยในระยะใกล้ ควรใช้สัญลักษณ์จุดเชื่อมต่อเข้ามาช่วย

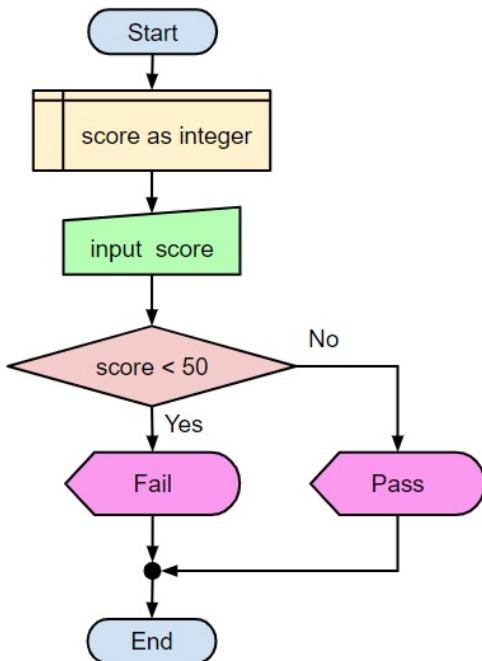
สำหรับเครื่องมือหรือซอฟต์แวร์ที่ใช้ในการเขียนผังงานโปรแกรม มีด้วยกันหลายตัว เช่น Microsoft Word, Microsoft Visio, Google Drawing หรือ Dia เป็นต้น

### ตัวอย่างที่ 1 ผังงานโปรแกรมของโปรแกรมคำนวณพื้นที่สามเหลี่ยม



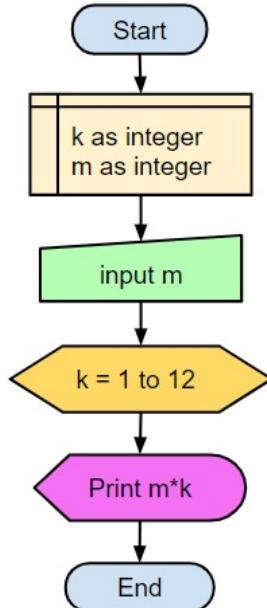
รูปที่ 1.11 แสดงผังงานโปรแกรมคำนวณพื้นที่สามเหลี่ยม

## ตัวอย่างที่ 2 ผังงานโปรแกรมตรวจสอบผลการเรียน



รูปที่ 1.12 แสดงผังงานโปรแกรมตรวจสอบผลการเรียน

## ตัวอย่างที่ 3 ผังงานโปรแกรมตารางสูตรคูณ



รูปที่ 1.13 แสดงผังงานโปรแกรมตารางสูตรคูณ

# กิจกรรมการเรียนรู้ หน่วยที่ 1

## กิจกรรม 1.1 กิจกรรมแบ่งกลุ่มเรียนรู้

ให้นักเรียนแบ่งกลุ่มเรียนรู้ร่วมกันกลุ่มละ 2-4 คน และจัดรวมเป็นทีม A B C ตามลำดับโดยมีทีมละ 4 กลุ่ม และตั้งชื่อกลุ่ม เป็น A1, A2, A3, A4 และ B1, B2, B3, B4 ตามลำดับ โดยระบุรายชื่อสมาชิกของกลุ่ม ดังนี้

กลุ่ม ..... มีสมาชิกดังนี้

1. ..... รหัสนักศึกษา .....
2. ..... รหัสนักศึกษา .....
3. ..... รหัสนักศึกษา .....
4. ..... รหัสนักศึกษา .....

## กิจกรรม 1.2 กิจกรรมวิเคราะห์งานโปรแกรม

ให้นักเรียนแต่ละกลุ่มซ่อมกันวิเคราะห์งานโปรแกรม ดังต่อไปนี้

1. โปรแกรมคำนวนพื้นที่สี่เหลี่ยมผืนผ้า
2. โปรแกรมคำนวนพื้นที่สี่เหลี่ยมจัตุรัส
3. โปรแกรมคำนวนพื้นที่สี่เหลี่ยมคางหมู
4. โปรแกรมคำนวนพื้นที่วงกลม

โดยดำเนินตามขั้นตอนดังตัวอย่าง

**ตัวอย่าง** การวิเคราะห์งานโปรแกรมคำนวนพื้นที่สามเหลี่ยม

### 1) ออกรูปแบบหน้าจอโปรแกรม

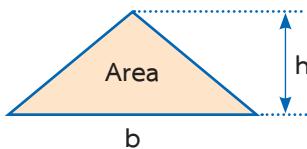
Program Area of Triangle. //แสดงชื่อโปรแกรม

Base = xx //รับข้อมูลความยาวฐาน

Height = xx //รับข้อมูลความสูงของสามเหลี่ยม

Area = xxxx.xx //แสดงผลลัพธ์ของโปรแกรม

## 2) วิเคราะห์งาน

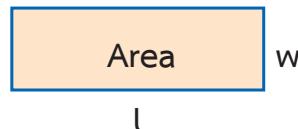


- 2.1 ข้อมูลนำเข้า คือ ความยาวฐาน และความสูงของสามเหลี่ยม
- 2.2 ตัวแปรที่ใช้ คือ  $b$  แทนความยาวฐานของสามเหลี่ยม กำหนดเป็นจำนวนเต็ม  
 $h$  แทนความสูงของสามเหลี่ยม กำหนดเป็นจำนวนเต็ม  
Area แทนพื้นที่สามเหลี่ยม เป็นแบบ ทศนิยม
- 2.3 การคำนวณผลลัพธ์ คือ  $\text{Area} = 0.5 * b * h$

งานโปรแกรม-1 โปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า (Rectangular)

- 1) หน้าจอแสดงผลของโปรแกรม

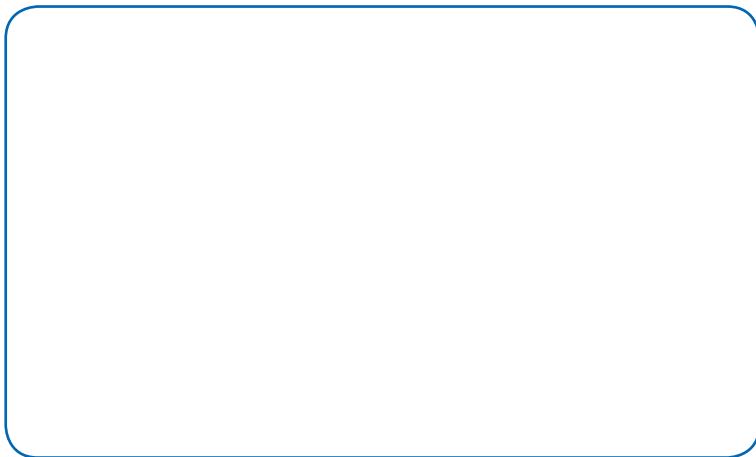
- 2) การวิเคราะห์งาน



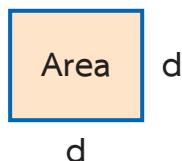
- 2.1 ข้อมูลนำเข้า คือ .....
- 2.2 ตัวแปรที่ใช้ คือ  $w$  แทน ..... กำหนดเป็นจำนวนเต็ม  
 $l$  แทน ..... กำหนดเป็นจำนวนเต็ม  
Area แทนพื้นที่สี่เหลี่ยมผืนผ้า กำหนดเป็น .....
- 2.3 การคำนวณผลลัพธ์ คือ  $\text{Area} = .....$

## งานโปรแกรม-2 โปรแกรมคำนวณพื้นที่สี่เหลี่ยมจัตุรัส (Square)

1) หน้าจอแสดงผลของโปรแกรม



2) การวิเคราะห์งาน



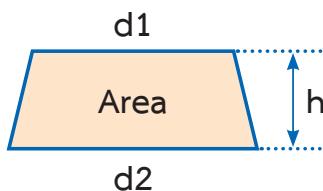
- 2.1 ข้อมูลนำเข้า คือ .....
- 2.2 ตัวแปรที่ใช้ คือ d แทน ..... กำหนดเป็นจำนวนเต็ม  
Area แทนพื้นที่สี่เหลี่ยมจัตุรัส กำหนดเป็น .....
- 2.3 การคำนวณผลลัพธ์ คือ Area = .....

### งานโปรแกรม-3 โปรแกรมคำนวณพื้นที่สี่เหลี่ยมคางหมู (Trapezoidal)

1) หน้าจอแสดงผลของโปรแกรม



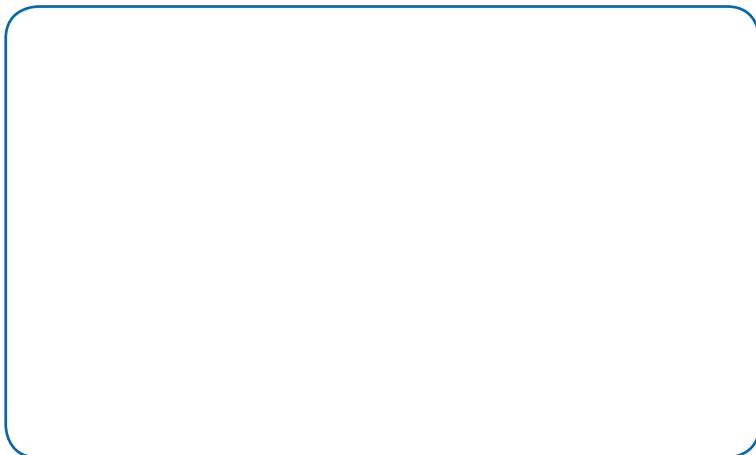
2) การวิเคราะห์งาน



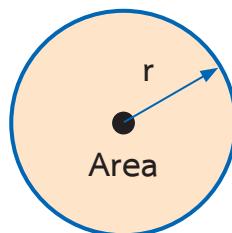
- 2.1 ข้อมูลนำเข้า คือ ความยาวของด้านคู่ขนาน และ .....
- 2.2 ตัวแปรที่ใช้ คือ
- d1 แทน ความยาวด้านคู่ขนานด้านที่ 1 กำหนดเป็นจำนวนเต็ม
  - d2 แทน ..... กำหนดเป็นจำนวนเต็ม
  - h แทน ..... กำหนดเป็นจำนวนเต็ม
  - Area แทนพื้นที่สี่คางหมู กำหนดเป็นแบบ .....
- 2.3 การคำนวณผลลัพธ์ คือ  $\text{Area} = \dots$

## งานโปรแกรม-4 โปรแกรมคำนวณพื้นที่วงกลม (Circle)

1) หน้าจอแสดงผลของโปรแกรม



2) การวิเคราะห์งาน

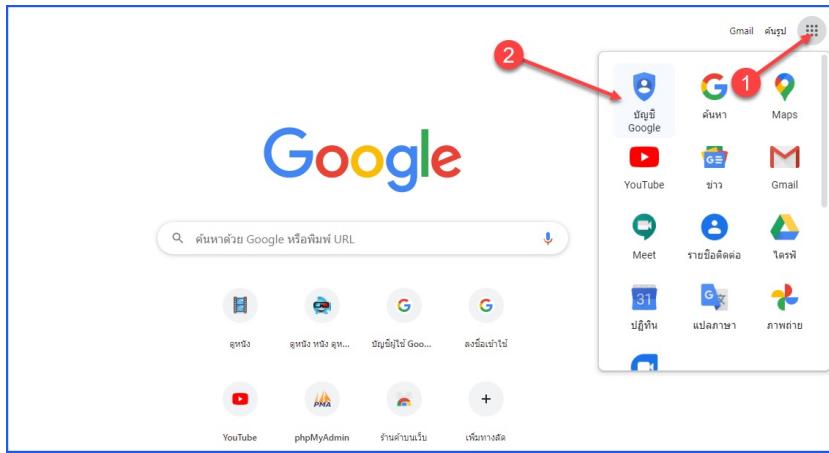


- 2.1 ข้อมูลนำเข้า คือ .....
- 2.2 ค่าคงที่ คือ  $\pi = 3.1416$
- 2.3 ตัวแปรที่ใช้ คือ
  - r แทน ..... กำหนดเป็นจำนวนเต็ม
  - Area แทนพื้นที่วงกลม กำหนดเป็น .....
- 2.4 การคำนวณผลลัพธ์ คือ  $Area = \dots$

### กิจกรรม 1.3 กิจกรรมเรียนรู้การเขียนผังงานด้วย Google Drawing

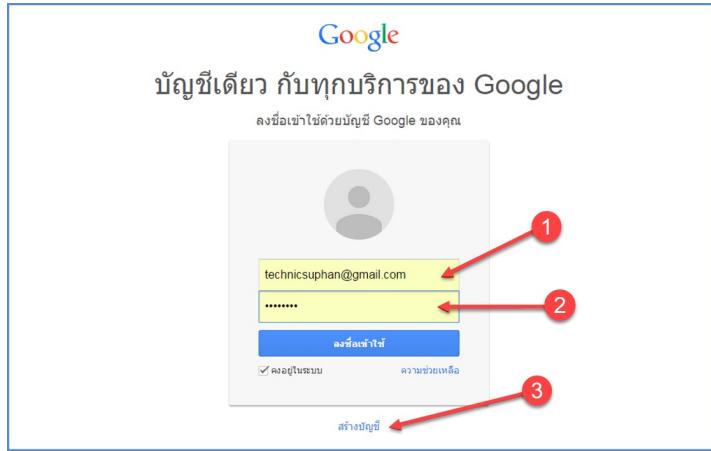
ให้นักเรียนฝึกการเขียนผังงานโปรแกรม (Program Flowchart) โดยใช้ Google Drawings โดยดำเนินการตามขั้นตอน ดังนี้

- 1) เข้าใช้งานอินเทอร์เน็ต
  - 2) เปิดโปรแกรม Google Chrome
  - 3) เข้าสู่เว็บไซต์ Google และเข้าสู่บัญชี Google ดังรูปที่ P1-1



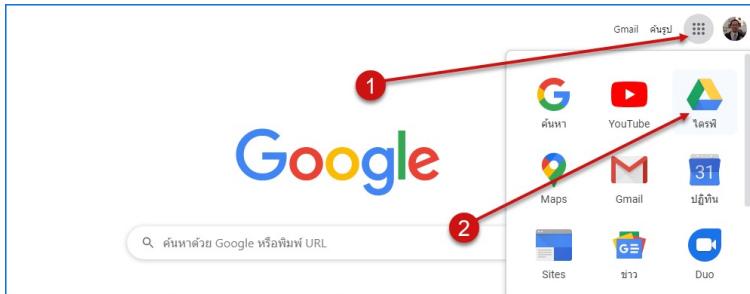
## ຮູບທີ່ P1-1

- 4) ป้อนชื่อบัญชีของ Google (1) และรหัสผ่าน (2) ในกรณีไม่มีรายการให้คลิกที่ข้อความ สร้างบัญชี (3) เพื่อสร้างบัญชีใหม่ ดังรูปที่ P1-2



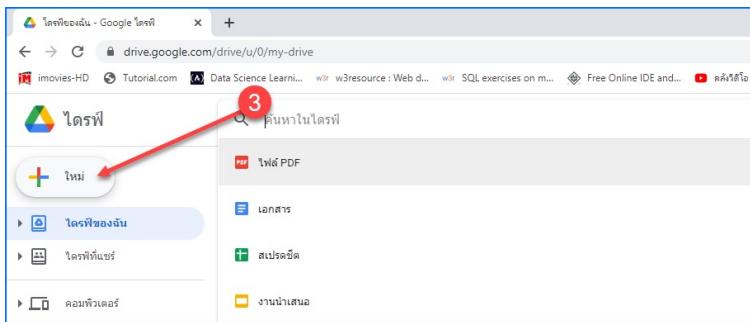
## ຮູບທີ່ P1-2

5) เมื่อเข้าสู่บัญชี Google ให้คลิกที่ปุ่ม App (1) และเลือก Google Drive (2) ดังรูปที่ P1-3



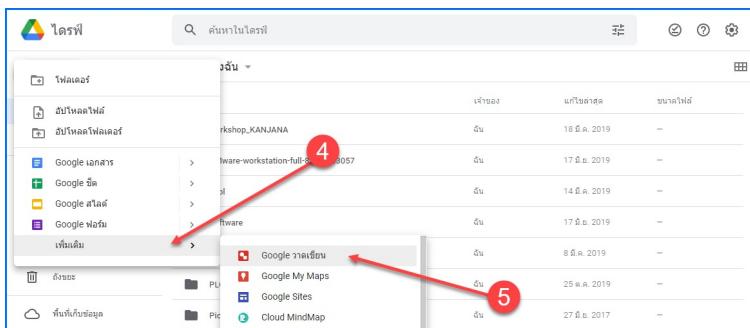
รูปที่ P1-3

6) จะปรากฏหน้าจอภาพ ดังรูปที่ P1-4 ให้คลิกที่ปุ่ม ใหม่ (New)



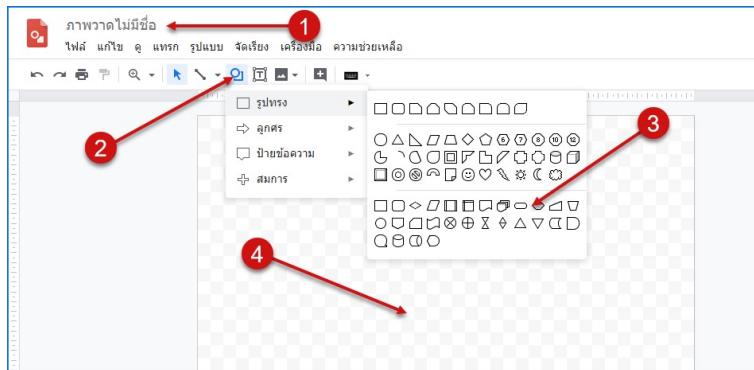
รูปที่ P1-4

7) สร้างงานด้วย Google Drawings โดยเลือกที่ Google วาดเขียน (Drawings) ดังรูปที่ P1-5



รูปที่ P1-5

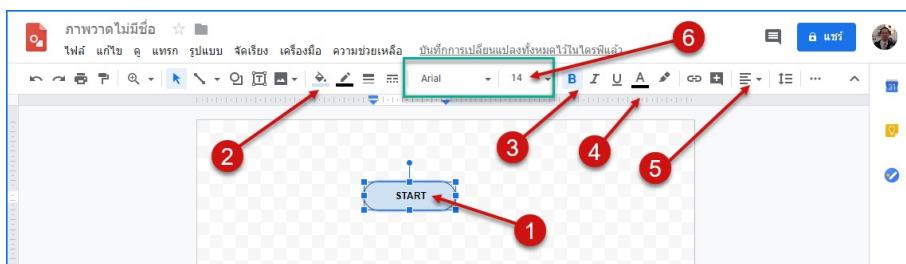
- 8) จะเข้าสู่หน้าจอโปรแกรมดังรูปที่ P1-6 โดย
- ส่วนที่ (1) ใช้กำหนดชื่องาน
  - ส่วนที่ (2) เป็นปุ่มเครื่องมือวาดรูปทรง
  - ส่วนที่ (3) เป็นรายการรูป平淡ที่ต้องการ
  - ส่วนที่ (4) คือ พื้นที่สร้างงาน



รูปที่ P1-6

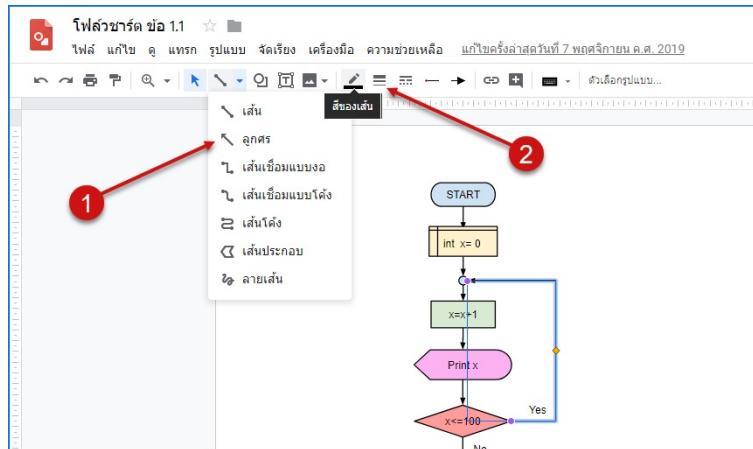
- 9) สร้างงาน โดยใช้รูป平淡ที่ต้องการวาดลงในพื้นที่สร้างงาน และใช้เครื่องมือเพื่อปรับแต่งส่วนต่าง ๆ ได้แก่

- (1) ดับเบิลคลิกเพื่อพิมพ์ข้อความในรูป平淡
- (2) ปรับเปลี่ยนสีพื้นของรูป平淡ได้ตามต้องการ
- (3) กำหนดลักษณะของตัวอักษร
- (4) กำหนดสีให้ตัวอักษร
- (5) จัดตำแหน่ง
- (6) เปลี่ยนแบบ และขนาดของตัวอักษร



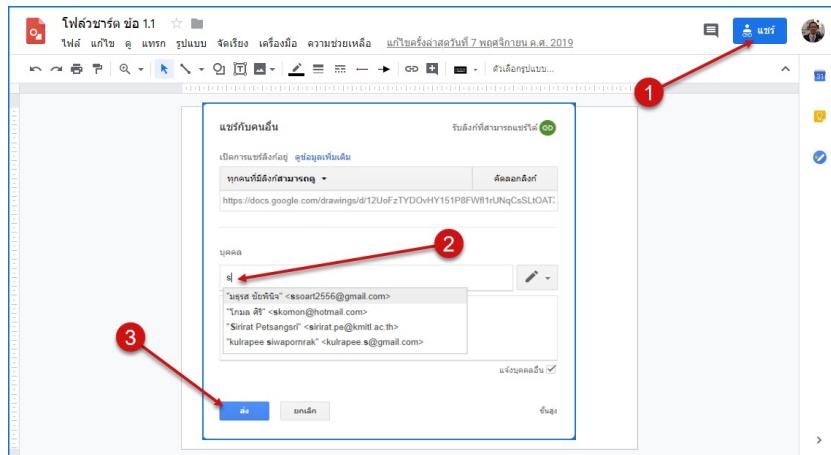
รูปที่ P1-7

10) วัดเส้นในผังงานเพื่อเขียนลำดับการทำงาน โดยใช้เครื่องมือ ลูกศร ในกรณีการเขียนเส้นแบบหักมุม ให้ใช้ เส้นเชื่อมแบบหักงอ โดยปรับขนาดเส้นและกำหนดสีเส้นได้ตามต้องการ ดังรูปที่ P1-8



รูปที่ P1-8

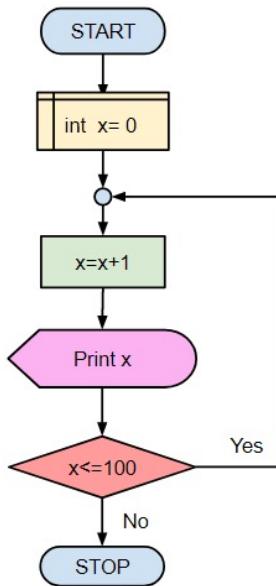
11) แชร์งานให้เพื่อนในกลุ่มช่วยกันวัดให้คลิกที่ปุ่ม [Share] ทางมุมบนด้านขวาจากนั้นกำหนดอีเมลของเพื่อนที่ทำงานร่วมกัน แล้วกดปุ่ม [ส่ง] ดังรูปที่ P1-9



รูปที่ P1-9

## กิจกรรม 1.4 กิจกรรมฝึกเขียนผังงานด้วย Google Drawing

ให้นักเรียนแต่ละคนฝึกเขียนผังงานโปรแกรมตามรูปที่ P1-10



รูปที่ P1-10

## กิจกรรม 1.5 กิจกรรมเขียนผังงานคำนวณพื้นที่ ด้วย Google Drawing

ให้แต่ละทีมแบ่งงานกันเขียนผังงานโปรแกรมต่อไปนี้

1. โปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า
2. โปรแกรมคำนวณพื้นที่สี่เหลี่ยมจัตุรัส
3. โปรแกรมคำนวณพื้นที่สี่เหลี่ยมคางหมู
4. โปรแกรมคำนวณพื้นที่วงกลม

โดยตกลงกันภายในทีมว่ากลุ่มใดจะทำหัวข้ออะไรกันนั้นให้แชร์ไฟล์งานให้เพื่อนภายในกลุ่มช่วยกันวัด เมื่อทำเสร็จให้ส่งไฟล์งานตั้งกล่าว เพื่อครุผู้สอนให้คะแนนและให้คำแนะนำเพิ่มเติม

## แบบฝึกหัดหน่วยที่ 1

**คำนี้จะ** ให้นักเรียนตอบคำถามต่อไปนี้

1. โปรแกรมคอมพิวเตอร์ คืออะไร และมีความสำคัญอย่างไร

.....  
.....

2. การเขียนโปรแกรมคืออะไร

.....

3. ภาษาคอมพิวเตอร์ คืออะไร แบ่งเป็นกี่แบบและมีภาษาอะไรบ้าง

.....  
.....

4. ตัวแปลภาษาคืออะไร แบ่งได้เป็นกี่ประเภทและแตกต่างกันอย่างไร

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

5. แนวคิดในการเขียนโปรแกรมคอมพิวเตอร์มีอย่างไรบ้าง

.....  
.....

6. ขั้นตอนการพัฒนาโปรแกรมในระดับต้นแบ่งเป็นขั้นตอนอะไรบ้าง

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

7. รหัสเทียม (Pseudo Code) และ ผังงานโปรแกรม (Flowchart) ต่างกันอย่างไร

.....  
.....

# ?

## แบบทดสอบ หน่วยที่ 1

# ?

คำชี้แจง จงเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว

### 1. โปรแกรมคอมพิวเตอร์ คืออะไร

- ก. ภาษาที่ใช้สื่องงานคอมพิวเตอร์  
ค. เครื่องมือที่ใช้จัดการคอมพิวเตอร์
- ข. ชุดคำสั่งที่ใช้สื่องงานคอมพิวเตอร์  
ง. การเขียนภาษาคอมพิวเตอร์

### 2. คอมพิวเตอร์แตกต่างจากอุปกรณ์อิเล็กทรอนิกส์อย่างไร

- ก. ทำงานตามคำสั่งของมนุษย์  
ค. เปลี่ยนแปลงการทำงานได้ตามชาร์ดแวร์
- ข. ใช้พลังงานน้อยกว่า  
ง. เปลี่ยนแปลงการทำงานจากซอฟต์แวร์

### 3. ข้อได้หมายถึงการเขียนโปรแกรมคอมพิวเตอร์

- ก. การเขียนคำสั่งให้คอมพิวเตอร์ทำงาน  
ข. กระบวนการแปลงข้อมูลให้เป็นสารสนเทศ  
ค. รวบรวมความรู้ความชำนาญเป็นฐานความรู้  
ง. การทำให้คอมพิวเตอร์มีความสามารถคล้ายมนุษย์

### 4. ภาษาที่เครื่องสามารถรับรู้ได้โดยตรงคือภาษาใด

- ก. ภาษาเครื่อง ข. ภาษาระดับต่ำ ค. ภาษาระดับสูง ง. ภาษา C

### 5. การแปลทีลະคำสั่ง คือ ลักษณะของตัวแปลภาษาแบบใด

- ก. แบบ interpreter ข. แบบ controller  
ค. แบบ compiler ง. แบบ commutater

### 6. ข้อได้คือลักษณะการเขียนโปรแกรมเชิงวัตถุ

- ก. Function Programming ข. Procedural Programming  
ค. Object Oriented Programming ง. Structure Programming

### 7. การทำงานแบบใดที่ไม่มีส่วนการตัดสินใจ

- ก. แบบตามลำดับ ข. แบบเลือกทำ  
ค. แบบวนทำซ้ำ ง. แบบทำงานตามเงื่อนไข

### 8. ข้อได้ไม่ใช่ขั้นตอนในการพัฒนาโปรแกรม

- ก. การวิเคราะห์ปัญหา ข. การออกแบบโปรแกรม  
ค. การบำรุงรักษาโปรแกรม ง. การใช้ทรัพยากรของโปรแกรม

## 9. ข้อใดคือลักษณะของ Flowchart

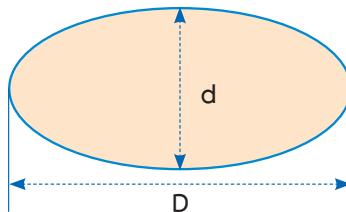
- ก. ผังงานที่ใช้คำสั่งอธิบายการทำงาน
- ข. ผังงานที่ใช้รหัสเทียมอธิบายขั้นตอนการทำงาน
- ค. การใช้สัญลักษณ์อธิบายขั้นตอนการทำงาน
- ง. แผนที่ความคิดของโปรแกรม

## 10. ข้อใดคือสัญลักษณ์การแสดงผลบนจอภาพของโฟลว์ชาร์ต

- ก. A parallelogram shape.
- ข. A rectangle shape.
- ค. An oval shape.
- ง. A diamond shape.

จากโจทย์ต่อไปนี้ ใช้ตอบคำถามข้อ 11 ถึง 15

โปรแกรมคำนวณพื้นที่ โดยถ้ากำหนดให้  $d$  = เส้นผ่าศูนย์กลางเส้นสั้น  $D$  = เส้นผ่านศูนย์กลางเส้นยาว และ  $\text{Area} = \text{พื้นที่วงรี}$  (ค่า  $\pi = 3.1416$ ) โดยพื้นที่วงรี = ผลบวกของเส้นผ่านศูนย์กลางคูณกับค่า  $\pi$  หารด้วย 4



## 11. ข้อใดน่าจะกำหนดเป็นค่าคงที่ของโปรแกรม

- ก.  $d$
- ข.  $D$
- ค.  $\pi$
- ง.  $\text{Area}$

## 12. ข้อใดคือตัวแปร input ของโปรแกรม

- ก.  $d$
- ข.  $D$
- ค.  $D$  และ  $d$
- ง.  $d, D$  และ  $\pi$

## 13. ข้อใดคือตัวแปร output ของโปรแกรม

- ก.  $d$
- ข.  $D$
- ค.  $\pi$
- ง.  $\text{Area}$

## 14. ข้อใดเขียนการคำนวณค่าผลลัพธ์ได้ถูกต้องที่สุด

- ก.  $\text{Area} = \pi * d * d$
- ข.  $\text{Area} = \pi * d * D$
- ค.  $\text{Area} = \pi * d + D / 4$
- ง.  $\text{Area} = \pi / 4 * (d + D)$

## 15. ถ้ากำหนดให้ input เป็นจำนวนเต็ม output ของโปรแกรม จะมีค่าอย่างไร

- ก. เป็นจำนวนเต็ม
- ข. เป็นทศนิยม
- ค. มีค่าไม่แน่นอน
- ง. จำนวนค่าไม่ได้