



คู่มือ Coding ภาษา

# PYTHON

ฉบับสมบูรณ์



- เริ่มต้นฝึกฝนอย่างถูกต้องง่ายๆ รวดเร็ว ด้วยหลักการและแบบฝึกหัดต่างๆ พร้อมคำอธิบายอย่างละเอียด
- ครอบคลุมพื้นฐานไปจนถึงการใช้งานระดับสูง อาทิ OOP, Database, Data Visualization, Data Science
- ปรับปรุงล่าสุด จาก Feedback ผู้อ่าน และประสบการณ์สอนมากกว่า 20 ปี



ไฟล์ตัวอย่างภายในเล่ม  
<https://serazu.com/>  
9786164872189

พศ.สุตา เรียร์มนตรี

<b>บทที่ 1</b>	<b>รู้จักกับภาษา Python (Introduction to Python)</b> .....	<b>1</b>
	แนะนำภาษา Python.....	1
	การทำงานของโปรแกรมภาษา Python.....	2
	แนะนำเครื่องมือสำหรับเขียนโปรแกรมภาษา Python.....	3
	การติดตั้งโปรแกรม Python.....	3
	การเรียกใช้งานโปรแกรม Python .....	5
	การติดตั้งและเรียกใช้งานโปรแกรม PyCharm.....	8
	แบบฝึกหัด .....	14
<b>บทที่ 2</b>	<b>เริ่มต้นเขียนโปรแกรมภาษา Python</b>	
	<b>(Introduction to Python Programming)</b> .....	<b>15</b>
	ขั้นตอนการเขียนโปรแกรมภาษา Python .....	15
	ขั้นที่ 1 วิเคราะห์ปัญหาของโปรแกรมที่ต้องการพัฒนา .....	16
	ขั้นที่ 2 ออกแบบการทำงานของโปรแกรม .....	16
	ขั้นที่ 3 เริ่มต้นสร้างและเขียนโปรแกรม .....	17
	ขั้นที่ 4 ทดสอบการทำงานของโปรแกรม .....	18
	การเขียนคำอธิบายโปรแกรม (Comment).....	19
	แบบฝึกหัด .....	20
<b>บทที่ 3</b>	<b>ตัวแปร ชนิดข้อมูล นิพจน์ และตัวดำเนินการ (Variable, Data Type,</b>	
	<b>Expression and Operator)</b> .....	<b>21</b>
	รู้จักกับตัวแปร (Variable).....	21
	กฎการตั้งชื่อตัวแปรในภาษา Python .....	21
	ชนิดของข้อมูล (Data Type) .....	22
	ข้อมูลชนิดตัวเลข (Number) .....	22
	ข้อมูลชนิดค่าความจริง (Boolean) .....	23
	ข้อมูลชนิด None.....	23

ข้อมูลแบบเรียงลำดับ (Sequence).....	24
ข้อมูลชนิดเซต (Set).....	25
ข้อมูลชนิดดิกชันนารี (Dictionary).....	26
การตรวจสอบชนิดของข้อมูล (Data Type Checking) .....	26
การแปลงชนิดของข้อมูล (Data Type Conversion).....	28
นิพจน์ (Expression).....	28
ตัวดำเนินการ (Operator).....	29
ตัวดำเนินการกำหนดค่า (Assignment Operator).....	29
ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator).....	31
ตัวดำเนินการเปรียบเทียบ (Comparison Operator) .....	32
ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator).....	33
ตัวดำเนินการระดับบิต (Bitwise Operator) .....	33
ตัวดำเนินการแบบเป็นสมาชิก (Membership Operator).....	36
ตัวดำเนินการแบบแสดงเอกลักษณ์ (Identity Operator).....	37
ลำดับความสำคัญของตัวดำเนินการ (Operator of Precedence).....	38
แบบฝึกหัด .....	40

## บทที่ 4 การแสดงผลและการรับข้อมูล (Data Output and Input) ..... 41

การแสดงผลข้อมูลด้วยฟังก์ชัน print().....	41
การใช้เครื่องหมาย + คั่นระหว่างข้อความ.....	41
การใช้เครื่องหมาย , คั่นระหว่างข้อความ.....	42
การใช้เครื่องหมาย * สำหรับแสดงผลข้อความที่ซ้ำกัน.....	43
การใช้ตัวอักขระพิเศษ.....	44
การใช้รหัสการแสดงผล.....	45
ฟังก์ชัน format().....	49
การรับข้อมูลด้วยฟังก์ชัน input() .....	51
การรับข้อมูล 1 ตัวแปร.....	51
การรับข้อมูลมากกว่า 1 ตัวแปร.....	52
แบบฝึกหัด .....	54

<b>บทที่ 5</b>	<b>คำสั่งควบคุมทิศทางการทำงานของโปรแกรม (Control Statement).....</b>	<b>55</b>
	คำสั่งควบคุมแบบตามลำดับ (Sequence Control Statement) .....	55
	คำสั่งควบคุมแบบมีทางเลือก (Selection Control Statement) .....	56
	คำสั่ง if : คำสั่งควบคุมให้โปรแกรมทำงานหรือไม่ทำงานในชุดคำสั่งที่กำหนด ...	57
	คำสั่ง if...else : คำสั่งควบคุมให้โปรแกรมเลือกทำงาน	
	ในทางเลือกใดทางเลือกหนึ่งจาก 2 ทางเลือก .....	59
	คำสั่ง if...elif...else : คำสั่งควบคุมให้โปรแกรมเลือกทำงาน	
	ในทางเลือกใดทางเลือกหนึ่งจากหลายทางเลือก .....	63
	คำสั่งควบคุมแบบทำซ้ำ (Iteration Control Statement) .....	66
	คำสั่ง while : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่ไม่แน่นอน .....	67
	คำสั่ง while in range : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่แน่นอน .....	68
	คำสั่ง while True : คำสั่งควบคุมแบบทำซ้ำโดยทำอย่างน้อย 1 รอบ .....	70
	คำสั่ง while else : คำสั่งควบคุมส่วนที่อยู่นอกเหนือการทำซ้ำ .....	72
	คำสั่ง for : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่แน่นอน .....	74
	คำสั่ง break : คำสั่งออกจากการทำงานในลูปทันที .....	79
	คำสั่ง continue : คำสั่งบังคับให้ข้ามไปทำงานรอบต่อไปทันที .....	81
	คำสั่ง pass : คำสั่งบังคับให้ผ่านไปทำงานในชุดคำสั่งถัดไป .....	82
	แบบฝึกหัด .....	84

<b>บทที่ 6</b>	<b>ข้อมูลชนิดเรียงลำดับ เซต และดิกชันนารี</b>	
	<b>(Sequence, Set and Dictionary Data Type).....</b>	<b>87</b>
	ข้อความ (String) .....	87
	การเข้าถึงข้อมูลชนิด String .....	88
	การเชื่อมต่อ และการทำซ้ำกับข้อมูลชนิด String.....	89
	การดำเนินการกับข้อมูลชนิด String .....	89
	ฟังก์ชันสำหรับข้อมูลชนิด String.....	90
	ลิสต์และทูเปิล (List and Tuple) .....	93
	การเข้าถึงข้อมูลชนิด List และ Tuple .....	94
	ฟังก์ชันสำหรับข้อมูลชนิด List และ Tuple .....	96
	เซต (Set) .....	100
	ฟังก์ชันสำหรับข้อมูลชนิด Set.....	101
	ข้อมูลชนิดดิกชันนารี (Dictionary) .....	104

การเข้าถึงข้อมูลชนิด Dictionary.....	104
ฟังก์ชันสำหรับข้อมูลชนิด Dictionary .....	105
แบบฝึกหัด .....	107

## บทที่ 7 ฟังก์ชัน (Function) ..... 109

ไลบรารีฟังก์ชัน (Library function) .....	109
Built in function .....	109
Module function .....	110
ฟังก์ชันในโมดูล datetime.....	112
ฟังก์ชันในโมดูล math .....	113
ฟังก์ชันที่เขียนขึ้นเอง (User defined function) .....	115
ตัวแปร global.....	116
ตัวแปร local .....	118
อาร์กิวเมนต์ (Argument) และพารามิเตอร์ (Parameter).....	119
รูปแบบการเขียนฟังก์ชัน.....	122
กรณีฟังก์ชันไม่มีการรับพารามิเตอร์และไม่มีการคืนค่า .....	122
กรณีที่ฟังก์ชันมีการรับพารามิเตอร์แต่ไม่มีการคืนค่า .....	124
กรณีที่ฟังก์ชันไม่มีการรับพารามิเตอร์แต่มีการคืนค่า .....	125
กรณีที่ฟังก์ชันมีการรับพารามิเตอร์และมีการคืนค่า.....	127
ฟังก์ชันโมดูล.....	132
ฟังก์ชันแลมด้า (Lambda function).....	134
แบบฝึกหัด .....	137

## บทที่ 8 การจัดการข้อผิดพลาด (Exception Handling)..... 139

รู้จักกับ Exception .....	139
ประเภทของ Exception.....	140
การใช้งานคำสั่ง try ... except .....	140
การใช้ else ร่วมกับคำสั่ง try ... except.....	142
การใช้ finally ร่วมกับคำสั่ง try ... except.....	148
แบบฝึกหัด .....	152

<b>บทที่ 9</b>	<b>Graphic User Interface (GUI).....</b>	<b>153</b>
	รู้จัก GUI Component.....	153
	การสร้าง GUI.....	153
	ประเภทของ GUI .....	154
	Window : หน้าต่างสำหรับจัดวางคอมโพเนนต์ .....	154
	Frame : เฟรมหรือกรอบแสดงผล.....	155
	Button : ปุ่ม และ PhotoImage : รูปภาพ.....	156
	Label และ Message : แสดงผลข้อความ .....	158
	Entry และ Text : รับและแสดงผลข้อความ.....	160
	Radiobutton : ตัวเลือกที่เลือกได้หนึ่งตัวเลือก .....	161
	Checkbox : ตัวเลือกที่เลือกได้มากกว่า 1 ตัวเลือก.....	163
	Listbox : รายการข้อมูล และ Scrollbar.....	166
	Combobox : รายการข้อมูลที่เลือกได้ 1 รายการ.....	167
	แบบฝึกหัด .....	169
<b>บทที่ 10</b>	<b>ตัวอย่างโปรแกรมประยุกต์ และ Event Handling .....</b>	<b>171</b>
	ตัวอย่างโปรแกรมประยุกต์ .....	171
	Event Handling.....	180
	การจัดการ command ด้วย Lambda function .....	187
	แบบฝึกหัด .....	191
<b>บทที่ 11</b>	<b>การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) .....</b>	<b>193</b>
	รู้จักคลาส (Class) และออบเจกต์ (Object) .....	193
	ประเภทของแอดทริบิวต์ .....	195
	ประเภทของเมธอด .....	197
	ระดับการเข้าถึงข้อมูล (Access Modifier) Public, Private และ Protected .....	201
	รู้จักและใช้งาน Encapsulation .....	204
	รู้จักและใช้งาน Inheritance และ Polymorphism .....	210
	แบบฝึกหัด .....	217

<b>บทที่ 12</b>	<b>ทำงานกับไฟล์ (File I/O).....</b>	<b>219</b>
	รู้จักกับ File.....	219
	การเขียนข้อมูลลงไฟล์.....	219
	การอ่านข้อมูลจากไฟล์.....	221
	ตัวอย่างการประยุกต์ใช้ไฟล์กับโปรแกรมจัดการอีเมล-คินหนังสือ.....	223
	โครงสร้างข้อมูล.....	223
	โครงสร้างโปรแกรม.....	224
	แบบฝึกหัด.....	246
<b>บทที่ 13</b>	<b>การเขียนโปรแกรมจัดการฐานข้อมูล (Database Programming).....</b>	<b>247</b>
	การเชื่อมต่อกับฐานข้อมูล SQLite.....	247
	ตัวอย่างการประยุกต์ใช้ฐานข้อมูล SQLite กับการจัดการข้อมูล	
	การโอนย้ายสินค้าในคลังสินค้า.....	248
	เริ่มต้นเขียนโปรแกรมกับฐานข้อมูล SQLite.....	248
	การเชื่อมต่อกับฐานข้อมูล Microsoft Access.....	266
	การติดตั้งโมดูล pyodbc และโมดูล pypyodbc.....	266
	การประยุกต์ใช้ฐานข้อมูล Microsoft Access กับการจัดการสมัครสมาชิก.....	268
	เริ่มต้นเขียนโปรแกรมกับฐานข้อมูล Microsoft Access.....	269
	การเชื่อมต่อกับฐานข้อมูล MySQL.....	283
	การติดตั้งโปรแกรม XAMPP.....	283
	การประยุกต์ใช้ฐานข้อมูล MySQL กับการจัดการร้านเบเกอรี่.....	287
	เริ่มต้นเขียนโปรแกรมกับฐานข้อมูล MySQL.....	287
	แบบฝึกหัด.....	304
<b>บทที่ 14</b>	<b>การสร้าง GUI Form ด้วย PyQt และ Qt Designer.....</b>	<b>305</b>
	การติดตั้งโมดูล PyQt และ Qt Designer.....	305
	การใช้งาน Qt Designer.....	306
	การแปลง GUI Form เป็นไฟล์ภาษา Python.....	309
	การเขียนคำสั่งเพื่อใช้งานโมดูล PyQt5.....	313
	ตัวอย่างโปรแกรมประยุกต์.....	315
	ตัวอย่างการประยุกต์ใช้ฐานข้อมูล SQLite กับการจัดการข้อมูลการรับสินค้าเข้าใน	
	คลังสินค้า.....	315
	แบบฝึกหัด.....	360

<b>บทที่ 15</b>	<b>การสร้างรายงานด้วย ReportLab .....</b>	<b>361</b>
	การสร้างรายงานด้วยคลาส Paragraph.....	361
	การสร้างรายงานด้วยคลาส Canvas.....	364
	การสร้างรายงานด้วยคลาส Table .....	369
	การสร้าง Chart ในรายงาน .....	373
	แบบฝึกหัด .....	379
<b>บทที่ 16</b>	<b>Python กับการประยุกต์ใช้ในงานด้านต่างๆ .....</b>	<b>381</b>
	การเขียนเว็บแอปพลิเคชันด้วยโมดูล Django .....	381
	ขั้นตอนการสร้างโปรเจกต์ Django.....	382
	การเขียนโปรแกรมเว็บแอปพลิเคชัน.....	385
	การส่งข้อความผ่านเครือข่ายด้วยโมดูล socket .....	392
	ขั้นตอนการทำงานระหว่าง Client-Server Mode.....	392
	การเขียนโปรแกรมระหว่าง Client-Server Mode.....	392
	การส่งจดหมายอิเล็กทรอนิกส์ Gmail ด้วยโมดูล smtplib .....	397
	แบบฝึกหัด .....	400
<b>บทที่ 17</b>	<b>Python กับงานด้าน Data Science .....</b>	<b>401</b>
	ความหมายของ Data Science .....	401
	ขั้นตอนของ Data Science.....	402
	การเขียนโปรแกรมจัดการข้อมูลด้วยโมดูล pandas.....	404
	รู้จักกับไฟล์ข้อมูล CSV.....	404
	ตัวอย่างการเขียนโปรแกรมจัดการข้อมูลด้วยโมดูล pandas.....	405
	รู้จักกับความซ้ำกันของข้อมูล และการแทนที่ข้อมูลที่สูญหาย.....	409
	ตัวอย่างการเขียนโปรแกรมจัดการความซ้ำกันของข้อมูล และข้อมูลสูญหาย ด้วยโมดูล pandas.....	410
	การวิเคราะห์และนำเสนอข้อมูลในรูปแบบของกราฟแผนภูมิด้วยโมดูล matplotlib .....	414
	ตัวอย่างการเขียนโปรแกรมวิเคราะห์และนำเสนอข้อมูลด้วยโมดูล matplotlib.....	414
	การจัดการข้อมูลด้วยโมดูล numpy .....	420
	การจัดการข้อมูลในรูปแบบอาร์เรย์ด้วยโมดูล numpy .....	420
	ตัวอย่างการเขียนโปรแกรมจัดการข้อมูลด้วยโมดูล numpy .....	422
	แบบฝึกหัด .....	424
	<b>Index .....</b>	<b>425</b>





python

# ฟังก์ชันโมดูล

จากที่กล่าวไว้ตอนต้นแล้วว่า โมดูลก็คือ ไฟล์ที่มีการรวบรวมฟังก์ชันหลายๆ ฟังก์ชันไว้ด้วยกัน ซึ่งผู้อ่านสามารถสร้างโดยการเขียนขึ้นเองได้ และสามารถเก็บไว้ใช้กับโปรแกรมใดก็ได้ตามที่ต้องการ เพื่อความเข้าใจให้ผู้อ่านศึกษาจากตัวอย่างโปรแกรมต่อไปนี้

**ตัวอย่างโปรแกรมที่ 7-11** โปรแกรมการคำนวณค่าแรง โดยเรียกใช้ฟังก์ชันที่เขียนไว้ในโมดูล wage มีโค้ดดังนี้

```

1      from wage import *
2
3      h = 50
4      r = 100
5      print("จำนวนชั่วโมงทำงาน เท่ากับ {:d} ชั่วโมง".format(h))
6      print("อัตราค่าแรง เท่ากับ {:.2f} บาทต่อชั่วโมง".format(r))
7      print("ค่าแรง 40 ชั่วโมง เท่ากับ {:.2f} บาท".format(calfirst(r)))
8      print("ค่าแรงที่เกินจาก 40 ชั่วโมง เท่ากับ {:.2f} บาท".format(calextra(h, r)))
9      print("ค่าแรงรวม เท่ากับ {:.2f} บาท".format(calwage(h, r)))

```

จากโปรแกรมที่ 7-11 อธิบายการทำงานของโปรแกรม ซึ่งเป็นส่วนของโปรแกรมหลักได้ดังนี้

- บรรทัดที่ 1     import ฟังก์ชันทั้งหมดจากโมดูล wage
- บรรทัดที่ 3-4   กำหนดจำนวนชั่วโมงทำงาน h เท่ากับ 50 ชั่วโมง และอัตราค่าแรง r เท่ากับ 100 บาทต่อชั่วโมง
- บรรทัดที่ 5-6   แสดงผลจำนวนชั่วโมงทำงาน h และอัตราค่าแรง r โดยจัดรูปแบบการแสดงผลด้วยฟังก์ชัน format()
- บรรทัดที่ 7     แสดงผลค่าแรง 40 ชั่วโมง โดยเรียกใช้ฟังก์ชัน calfirst() ที่อยู่ในโมดูล wage() และจัดรูปแบบการแสดงผลด้วยฟังก์ชัน format()
- บรรทัดที่ 8     แสดงผลค่าแรงที่เกินจาก 40 ชั่วโมง โดยเรียกใช้ฟังก์ชัน calextra() ที่อยู่ในโมดูล wage() และจัดรูปแบบการแสดงผลด้วยฟังก์ชัน format()
- บรรทัดที่ 9     แสดงผลค่าแรงรวม โดยเรียกใช้ฟังก์ชัน calwage() ที่อยู่ในโมดูล wage() และจัดรูปแบบการแสดงผลด้วยฟังก์ชัน format()

# ฟังก์ชันแลมด้า (Lambda function)

ฟังก์ชันแลมด้า เป็นฟังก์ชันที่ผู้อ่านสามารถสร้างขึ้นมาเพื่อใช้งานได้โดยไม่ต้องระบุชื่อ (Anonymous functions) หรืออาจเรียกว่า **ฟังก์ชันนิรนาม** โดยมากจะประกาศเป็นฟังก์ชันเล็กๆ ที่สร้างขึ้นเพื่องานประมวลผลง่ายๆ ไม่ซับซ้อน

การทำงานของฟังก์ชันแลมด้าจะเหมือนฟังก์ชันทั่วไปคือ ผู้อ่านสามารถสร้างฟังก์ชันให้รับค่าพารามิเตอร์ก็ได้ แต่จะต้องประกอบด้วย 1 นิพจน์การทำงาน หรือ 1 Expression เท่านั้น และไม่ต้องกำหนดให้คืนค่า เนื่องจากจะมีการคืนค่าโดยอัตโนมัติ ซึ่งมีรูปแบบการใช้งานดังนี้

```
varName = lambda argument_1 [, argument_2, ..., argument_n]: expression
```

โดยที่	<b>varName</b>	เป็นชื่อตัวแปรรับค่าที่คืนมาจากฟังก์ชัน	
	<b>argument_1,</b> <b>argument_2,</b> <b>argument_n</b>	} เป็นค่าข้อมูลที่ส่งให้กับฟังก์ชัน	
	<b>expression</b>		เป็นนิพจน์สำหรับการประมวลผลของฟังก์ชัน

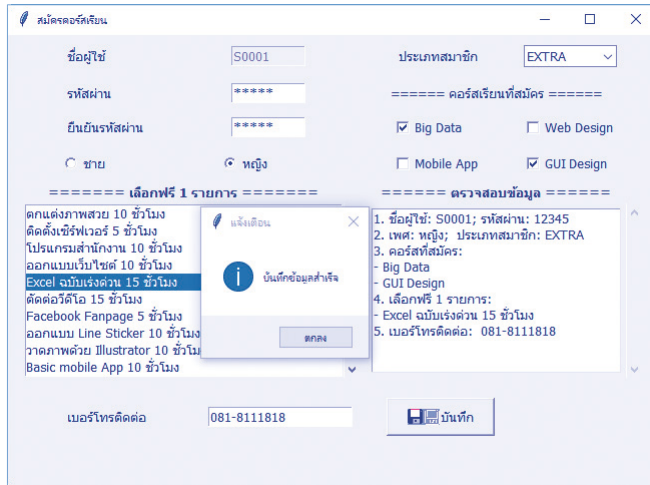
ตัวอย่างเช่น

```
a = lambda x, n: x * n  
print(a(3, 5))
```

```
#ผลที่ได้คือ  
15
```

เราสามารถนำฟังก์ชันแลมด้ามาใช้ร่วมกับฟังก์ชันที่สร้างขึ้นเอง เพื่อรวมฟังก์ชันที่มีการประมวลผลคล้ายกันเป็นฟังก์ชันเดียว แทนการสร้างหลายฟังก์ชันซ้ำๆ กัน เพื่อความเข้าใจ ผู้อ่านสามารถศึกษาจากตัวอย่างโปรแกรมต่อไปนี้

## ผลการทำงาน เมื่อคลิกปุ่ม "บันทึก"



จากผลลัพธ์ของโปรแกรมข้างต้น เราสามารถเขียนโค้ดได้ดังนี้

```
1 from tkinter import *
2 from tkinter.ttk import Combobox
3 from tkinter import messagebox
4
5 root = Tk()
6 root.title("สมัครคอร์สเรียน")
7 root.geometry("650x450")
8 root.option_add("*font", 'tahoma 10')
9 frame = Frame(root)
10 frame.pack()
11
12 idlabel = Label(frame, text="ชื่อผู้ใช้", width=12, anchor=W)
13 idlabel.grid(column=0, row=0, padx=5, pady=10)
14 pwlabel = Label(frame, text="รหัสผ่าน", width=12, anchor=W)
15 pwlabel.grid(column=0, row=1, padx=5, pady=5)
16 cpwlabel = Label(frame, text="ยืนยันรหัสผ่าน", width=12, anchor=W)
17 cpwlabel.grid(column=0, row=2, padx=5, pady=5)
18 tellabel = Label(frame, text="เบอร์โทรศัพท์", width=12, anchor=W)
19 tellabel.grid(column=0, row=6, padx=20, pady=20)
20
21 envar1 = StringVar(value="S0001")
22 entry1 = Entry(frame, textvariable=envar1, width=10, state=DISABLED);
23 entry1.grid(column=1, row=0, padx=5, pady=5)
24 envar2 = StringVar(value="12345")
25 entry2 = Entry(frame, textvariable=envar2, show="*", width=10);
26 entry2.grid(column=1, row=1, padx=5, pady=5)
27 envar3 = StringVar(value="12345")
```

## 2. จัดการผ่านการผูกเหตุการณ์เข้ากับคอมโพเนนต์ที่ต้องการตรวจจับเหตุการณ์ เช่น

- กดคีย์ใดๆ บนแป้นพิมพ์ที่ Entry, Text หรือ Button
- คลิกเลือกรายการที่ Listbox

จากตัวอย่างโปรแกรมข้างต้น ผู้อ่านสามารถจัดการผ่านการผูกเหตุการณ์เข้ากับคอมโพเนนต์ที่ต้องการได้ดังนี้

- บรรทัดที่ 40 ผูกเหตุการณ์กับ entry4 ด้วยคำสั่ง `entry4.bind('<Return>', entry4_keypress)` เมื่อมีการกดปุ่ม `Enter` หรือ `Return` บนแป้นพิมพ์ โปรแกรมจะเรียกใช้งานฟังก์ชัน `entry4_keypress(event)` ตามคำสั่งในบรรทัดที่ 34-38
  - บรรทัดที่ 34 สร้างฟังก์ชันชื่อ `entry4_keypress(event)`
  - บรรทัดที่ 35 กำหนดตัวแปรเบอร์โทรติดต่อ `tel` เป็นประเภท `global`
  - บรรทัดที่ 36 กำหนดค่า `tel` ให้เท่ากับค่าในช่อง `entry4` ด้วยคำสั่ง `event.widget.get()`
  - บรรทัดที่ 37 เรียกใช้งานฟังก์ชัน `checkdata`
  - บรรทัดที่ 38 กำหนดค่า `focus_set()` ให้กับปุ่ม `button1`
- บรรทัดที่ 70 ผูกเหตุการณ์กับ `listbox1` ด้วยคำสั่ง `listbox1.bind('<<ListboxSelect>>', listbox1_select)` เมื่อมีการคลิกเลือกรายการ โปรแกรมจะเรียกใช้งานฟังก์ชัน `listbox1_select(event)` ตามคำสั่งในบรรทัดที่ 64-68
  - บรรทัดที่ 64 สร้างฟังก์ชันชื่อ `listbox1_select(event)`
  - บรรทัดที่ 65-66 กำหนดค่า `w` และ `data` เพื่อเข้าถึงข้อมูลที่ผู้อ่านคลิกเลือกใน `listbox1`
  - บรรทัดที่ 67 กำหนดตัวแปรรายการที่เลือกฟรี `freeitem` เป็นประเภท `global`
  - บรรทัดที่ 68 กำหนดค่าให้กับตัวแปร `freeitem`

สำหรับรายละเอียดการตรวจจับเหตุการณ์ที่เกิดกับคอมโพเนนต์ต่างๆ ผู้อ่านสามารถศึกษารายละเอียดเพิ่มเติมได้ที่ <https://effbot.org/tkinterbook/tkinter-events-and-bindings.htm>

นอกจากการจัดการเหตุการณ์ในรูปแบบของการเรียกใช้ฟังก์ชัน ที่สร้างขึ้นเองให้ทำงานตามที่กล่าวมาแล้ว ผู้อ่านยังสามารถจัดการเหตุการณ์ของออปชัน `command` โดยใช้ฟังก์ชันแลมด้าได้ด้วย ซึ่งผู้อ่านจะได้เรียนรู้ในหัวข้อถัดไป

# รู้จักและใช้งาน Inheritance และ Polymorphism

**Inheritance** เป็นคุณสมบัติที่ว่า คลาสสามารถสืบทอดแอตทริบิวต์และเมธอดจากคลาสแม่ ไปสู่คลาสลูกได้ นั่นคือ คลาสที่ถูกสืบทอดมานั้น จะมีแอตทริบิวต์และเมธอดเหมือนกับคลาสแม่ อีกทั้งยังสามารถพัฒนาต่อ โดยเพิ่มเติมและแก้ไขแอตทริบิวต์และเมธอดได้ ซึ่งเรียกเมธอดนี้ว่า

## Overriding Method

**Polymorphism** เป็นคุณสมบัติที่ว่า ความสามารถของคลาสมีได้หลายรูปแบบ นั่นคือ การระบุ ชนิดและจำนวนข้อมูลที่ใช้ในการทำงานของเมธอดแตกต่างกัน ทำให้オブジェクトมีวิธีการทำงานที่ แตกต่างกันไป เรียกเมธอดนี้ว่า **Overloading Method**

เพื่อความเข้าใจ ให้ผู้อ่านศึกษาการใช้งานจากตัวอย่างโปรแกรมต่อไปนี้

### ตัวอย่างโปรแกรมที่ 11-8 โปรแกรมการใช้แนวคิด Inheritance และ Polymorphism

ตัวอย่างนี้เป็นโปรแกรมที่ใช้แนวคิด Inheritance และ Polymorphism กับการคำนวณเงิน ค่าภาษีและค่าขนส่ง ที่มีขั้นตอนและวิธีการคำนวณภายใต้เมธอดที่มีการสืบทอดคุณสมบัติจาก คลาสแม่ไปยังคลาสลูก และแสดงผลลัพธ์ของโปรแกรมดังนี้

ข้อมูลสินค้า

ประเภทสินค้า: Prime

ราคาสินค้า: 10000

เงินเพิ่ม: 1000

คำนวณ    ล้างข้อมูล

ค่าภาษี = 700.00 บาท

ค่าขนส่ง = 2,000.00 บาท

ผลการทำงาน กรณีประเภทสินค้าเท่ากับ Prime

ข้อมูลสินค้า

ประเภทสินค้า: Defect

ราคาสินค้า: 10000

เงินเพิ่ม: 0.0

คำนวณ    ล้างข้อมูล

ค่าภาษี = 250.00 บาท

ค่าขนส่ง = 500.00 บาท

ผลการทำงาน กรณีประเภทสินค้าเท่ากับ Defect

# ตัวอย่างการประยุกต์ใช้ไฟล์กับโปรแกรมจัดการยืม-คืนหนังสือ

สำหรับการเขียนโปรแกรมเพื่อจัดการงานที่มีข้อมูลไม่ซับซ้อน ผู้อ่านสามารถนำไฟล์มาใช้ในการจัดเก็บข้อมูลได้ เนื่องจากไฟล์ต้องการขนาดของพื้นที่ในหน่วยความจำไม่มาก อีกทั้งยังไม่มีปัญหาในการจัดหาโปรแกรมฐานข้อมูลที่ถูกต้องตามลิขสิทธิ์ อีกทั้งถ้าในอนาคตต้องการพัฒนาเป็นการใช้โปรแกรมฐานข้อมูล ก็สามารถนำเข้าสู่ข้อมูลจากไฟล์ไปยังโปรแกรมฐานข้อมูลต่างๆ ได้

ตัวอย่างโปรแกรมต่อไป จะเป็นการจัดการยืม-คืนหนังสือ โดยใช้ไฟล์ในการจัดเก็บข้อมูล และใช้การออกแบบส่วนติดต่อผู้ใช้แบบกราฟิก เพื่อให้โปรแกรมมีความน่าสนใจและใช้งานง่าย ซึ่งประกอบด้วย

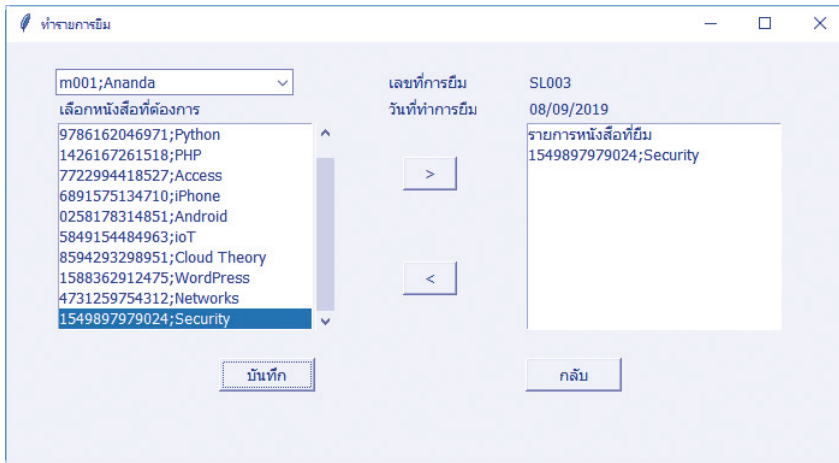
- 1) การเพิ่มข้อมูลหนังสือ
- 2) การทำรายการยืม
- 3) การทำรายการคืน

## โครงสร้างข้อมูล

ตัวอย่างนี้เป็นการแสดงผลข้อมูลที่จัดเก็บด้วยไฟล์ ซึ่งประกอบด้วยรายละเอียดข้อมูลดังนี้

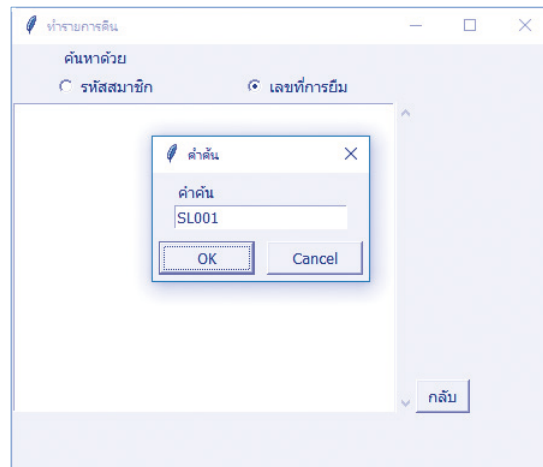
- ไฟล์ข้อมูลหนังสือ (book.txt) ประกอบด้วยข้อมูล
  1. เลขบาร์โค้ด
  2. ชื่อหนังสือ
  3. สถานะหนังสือ
- ไฟล์ข้อมูลสมาชิก (member.txt) ประกอบด้วยข้อมูล
  1. รหัสสมาชิก
  2. ชื่อสมาชิก
- ไฟล์รายการยืมคืน (borrow.txt) ประกอบด้วยข้อมูล
  1. เลขบาร์โค้ด
  2. ชื่อหนังสือ
  3. รหัสสมาชิก
  4. สถานะการยืม

- ผลการทำงาน เมื่อคลิกปุ่ม "ทำรายการยืม"



เมื่อผู้อ่านคลิกเลือกรายชื่อสมาชิก และเลือกรายการข้อมูลหนังสือที่ต้องการ แล้วคลิกปุ่ม "บันทึก" ข้อมูลก็จะถูกเขียนลงไฟล์ borrow.txt และหน้าจอจะถูกล้างข้อมูลพร้อมกับแสดงเลขที่รายการยืมใหม่ เพื่อรอการทำรายการยืมต่อไป

- ผลการทำงาน เมื่อคลิกปุ่ม "ทำรายการคืน"





# รู้จักและใช้งาน Inheritance และ Polymorphism

**Inheritance** เป็นคุณสมบัติที่ว่า คลาสสามารถสืบทอดแอตทริบิวต์และเมธอดจากคลาสแม่ไปสู่คลาสลูกได้ นั่นคือ คลาสที่ถูกสืบทอดมานั้น จะมีแอตทริบิวต์และเมธอดเหมือนกับคลาสแม่ อีกทั้งยังสามารถพัฒนาต่อ โดยเพิ่มเติมและแก้ไขแอตทริบิวต์และเมธอดได้ ซึ่งเรียกเมธอดนี้ว่า

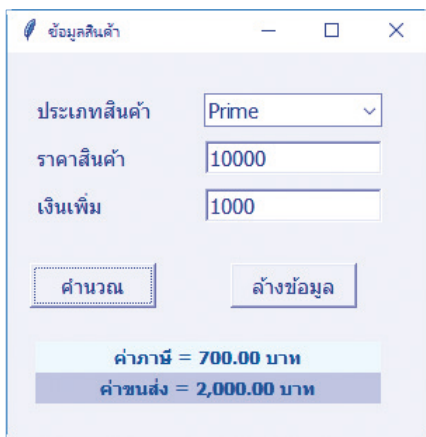
## Overriding Method

**Polymorphism** เป็นคุณสมบัติที่ว่า ความสามารถของคลาสมีได้หลายรูปแบบ นั่นคือ การระบุชนิดและจำนวนข้อมูลที่ใช้ในการทำงานของเมธอดแตกต่างกัน ทำให้オブジェクトมีวิธีการทำงานที่แตกต่างกัน เรียกเมธอดนี้ว่า **Overloading Method**

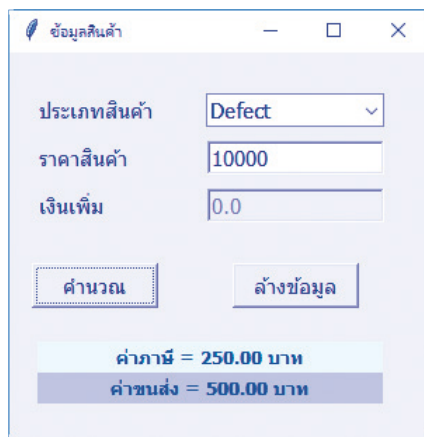
เพื่อความเข้าใจ ให้ผู้อ่านศึกษาการใช้งานจากตัวอย่างโปรแกรมต่อไปนี้

### ตัวอย่างโปรแกรมที่ 11-8 โปรแกรมการใช้แนวคิด Inheritance และ Polymorphism

ตัวอย่างนี้เป็นโปรแกรมที่ใช้แนวคิด Inheritance และ Polymorphism กับการคำนวณเงินค่าภาษีและค่าขนส่ง ที่มีขั้นตอนและวิธีการคำนวณภายใต้เมธอดที่มีการสืบทอดคุณสมบัติจากคลาสแม่ไปยังคลาสลูก และแสดงผลลัพธ์ของโปรแกรมดังนี้



ผลการทำงาน กรณีประเภทสินค้าเท่ากับ Prime



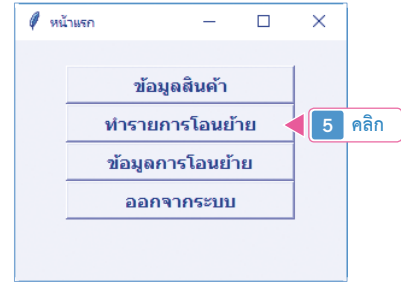
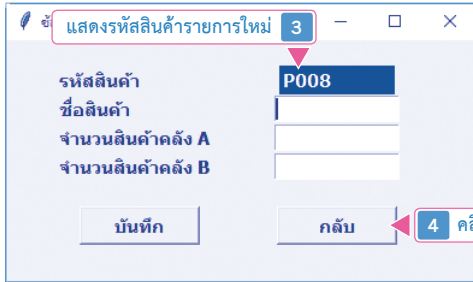
ผลการทำงาน กรณีประเภทสินค้าเท่ากับ Defect

- ฟอรัม ข้อมูลสินค้า

- ฟอรัม ทำรายการโอนย้าย

- ฟอรัม ข้อมูลการโอนย้าย

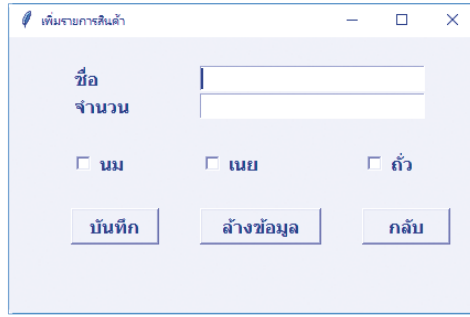
3. โปรแกรมจะบันทึกข้อมูลลงตาราง product และเคลียร์ข้อมูลบนหน้าจอพร้อมทั้งแสดงข้อมูลรหัสสินค้ารายการใหม่
4. เมื่อคลิกปุ่ม **กลับ** โปรแกรมจะแสดงหน้าจอ หน้าแรก
5. เมื่อคลิกปุ่ม **ทำรายการโอนย้าย** โปรแกรมจะแสดงหน้าจอ ทำรายการโอนย้าย



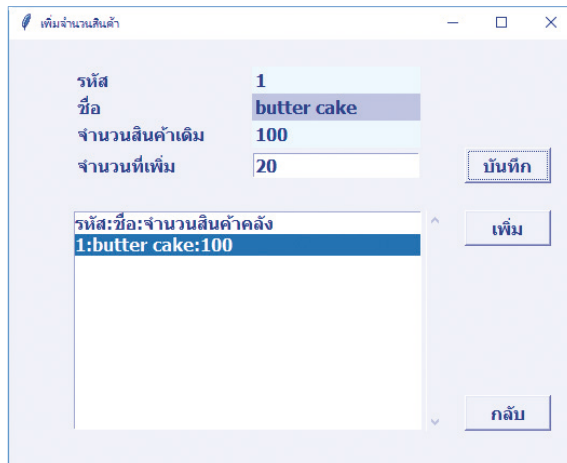
6. หน้าจอ ทำรายการโอนย้าย
  - 6.1 คลิกเลือกรายการ P002
  - 6.2 คลิกปุ่ม **...**
  - 6.3 ที่ Radiobutton คลิกเลือก A -> B
  - 6.4 ป้อนข้อมูลจำนวนที่โอนย้าย
  - 6.5 คลิกปุ่ม **บันทึก**



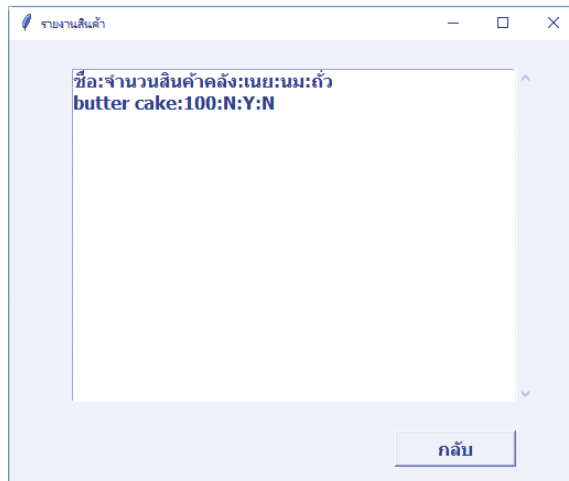
- เพิ่มรายการสินค้า (cakeform.py)



- เพิ่มจำนวนสินค้า (addform.py)



- รายงานสินค้า (listform.py)



## 1.2 โปรแกรมจัดการสวนติดต่อผู้ใช้ เพิ่มรายการสินค้า (cakeform.py) มีโค้ดดังนี้

```
1 from tkinter import *
2 from cakesDB import *
3 from tkinter import messagebox
4
5 def cakedata():
6
7     def insertproduct():
8         pname = nameentry.get()
9         pqty = int(qtyentry.get())
10        if var1.get() == 0:
11            check1 = 'N'
12        else:
13            check1 = 'Y'
14        if var2.get() == 0:
15            check2 = 'N'
16        else:
17            check2 = 'Y'
18        if var3.get() == 0:
19            check3 = 'N'
20        else:
21            check3 = 'Y'
22        productname = selectproductname(pname)
23        if productname:
24            messagebox.showinfo("แจ้งเตือน", "ชื่อสินค้านี้มีอยู่แล้ว")
25            nameentry.focus_set()
26        else:
27            result = insertproducttable(pname, pqty, check1, check2,
28                                       check3)
29            if result:
30                messagebox.showinfo("แจ้งเตือน", "บันทึกข้อมูลสำเร็จ")
31                cleardata()
32
33    def cleardata():
34        nametext.set("")
35        qtytext.set("")
36        var1.set("0")
37        var2.set("0")
38        var3.set("0")
39        nameentry.focus_set()
40
41    def root2exit():
42        root2.destroy()
43
44    root2 = Tk()
45    root2.title('เพิ่มรายการสินค้า')
```

จากโปรแกรม cakeform.py อธิบายการทำงานของโปรแกรมได้ดังนี้

บรรทัดที่ 1-3 อิมพอร์ตโมดูล tkinter และ cakesDB

บรรทัดที่ 5-91 สร้างฟังก์ชัน cakedata() เพื่อจัดการหน้าจอ เพิ่มรายการสินค้า ประกอบด้วย

- บรรทัดที่ 7-31 สร้างฟังก์ชัน insertproduct() เพื่อ
  - 1) เรียกใช้ฟังก์ชัน selectproductname(productname) ในโมดูล cakesDB ให้ตรวจสอบชื่อสินค้าซ้ำ
  - 2) เรียกใช้ฟังก์ชัน insertproducttable() ในโมดูล cakesDB ให้ insert ข้อมูลสินค้า
- บรรทัดที่ 33-39 สร้างฟังก์ชัน cleardata() เพื่อล้างข้อมูลบนหน้าจอ เพิ่มรายการสินค้า
- บรรทัดที่ 41-42 สร้างฟังก์ชัน root2exit() เพื่อปิดหน้าจอ เพิ่มจำนวนสินค้า ด้วยฟังก์ชัน destroy()
- บรรทัดที่ 44-91 เป็นการสร้างหน้าจอสำหรับรับข้อมูลเพื่อใช้ในการเพิ่มรายการสินค้า

### 1.3 โปรแกรมจัดการสวนติดต่อผู้ใช้ เพิ่มจำนวนสินค้า (addform.py) มีโค้ดดังนี้

```
1 from tkinter import *
2 from cakesDB import *
3
4 def adddata():
5
6     def updateproduct(i):
7         data = listbox1.get(i)
8         if data != 'รหัส:ชื่อ:จำนวนสินค้าคลัง':
9             data = data.split(":",3)
10            identry['text'] = data[0]
11            pid = int(data[0])
12            nameentry['text'] = data[1]
13            qtyentry['text'] = data[2]
14            pqty = int(data[2])
15            addentry.focus_set()
16            pqty = pqty + int(addentry.get())
17            updateproducttable(pid, pqty)
18            getproductlist()
19            identry['text'] = ""
20            nameentry['text'] = ""
21            qtyentry['text'] = ""
22            addtext.set("")
23            listbox1.focus_set()
```

```

90 Form.setWindowTitle(_translate("Form", "Form: insert"))
91 self.idlabel.setText(_translate("Form", "ID"))
92 self.savebutton.setText(_translate("Form", "save"))
93 self.savebutton.setShortcut(_translate("Form", "Enter"))
94 self.resetbutton.setText(_translate("Form", "reset"))
95 self.emaillabel.setText(_translate("Form", "E-mail"))
96 self.namelabel.setText(_translate("Form", "Name"))

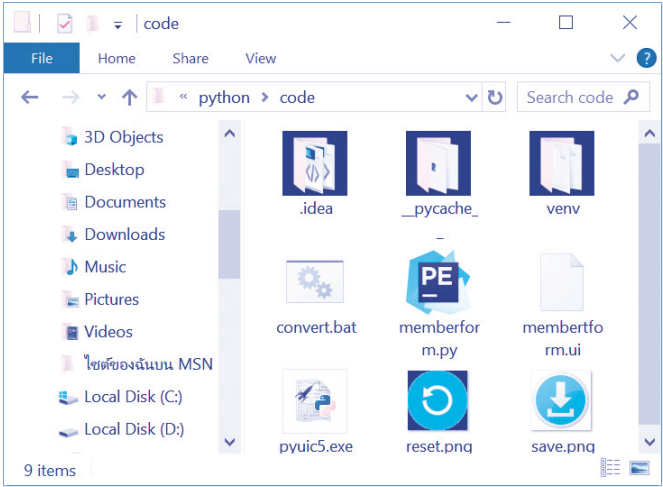
```

หลังจากที่เราแปลงไฟล์ memberform.ui ให้เป็น memberform.py แล้ว จะเป็นการเขียนคำสั่งใช้งานโมดูล PyQt5 เพื่อทำให้ GUI Form สามารถทำงานได้ ซึ่งผู้อ่านจะได้ศึกษาในหัวข้อถัดไป

# การเขียนคำสั่งเพื่อใช้งานโมดูล PyQt5

ในการเขียนคำสั่งเพื่อใช้งานโมดูล PyQt5 มีขั้นตอนดังนี้

1. คัดลอกโฟลเดอร์ PyQt5 ที่จัดเก็บอยู่ที่โฟลเดอร์ `..\python\code\venv\Lib\site-packages\` มาไว้ที่ `..\python\code\`



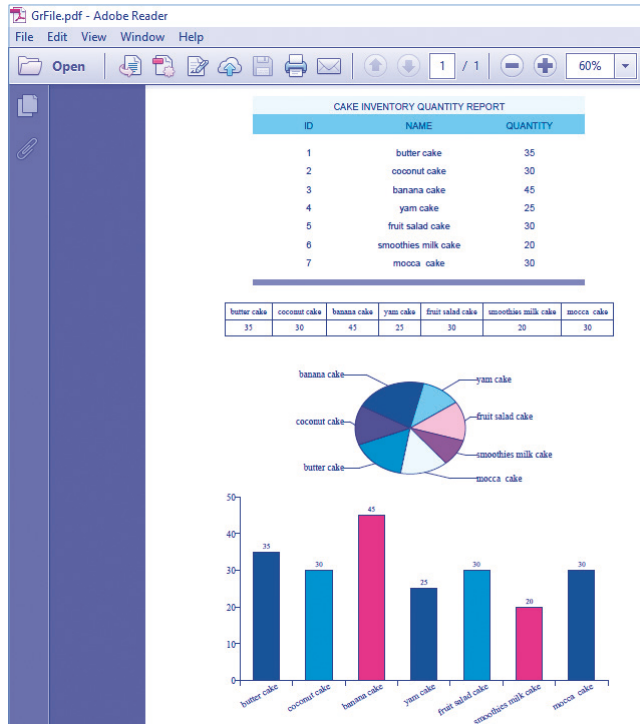
2. เขียนโปรแกรม `run_memberform.py` เพื่อสั่งให้ไฟล์ `memberform.py` ทำงาน ซึ่งมีโค้ดดังนี้

```

1 from PyQt5.Qtwidgets import Qwidget, QApplication, QMainWindow
2 import memberform
3 import sys
4
5
6 class MyApp(QMainWindow):
7     def __init__(self, parent=None):
8         Qwidget.__init__(self, parent)
9         self.ui = memberform.Ui_Form()

```

### 3. ข้อมูลถูกนำไปสร้างเป็นไฟล์ GrFile.pdf ซึ่งแสดงผลดังนี้



## สรุปท้ายบท

ในบทนี้ผู้อ่านได้เรียนรู้การสร้างรายงานด้วยโมดูล ReportLab ที่สามารถติดตั้งได้โดยง่าย และสามารถใช้งานสร้างรายงานเพื่อนำเสนอข้อมูลได้จากหลายแหล่ง ทั้งที่เป็นข้อมูลจากฐานข้อมูล ไฟล์ หรือข้อมูลบนหน้าจอ รูปแบบการสร้างรายงานก็ทำได้ทั้งแบบ Paragraph ที่นำข้อมูลไปจัดวางด้วยค่ารูปแบบต่างๆ แบบ Canvas ที่เป็นกราวด์วัตถุอ้างอิงพิกัดตำแหน่ง x, y แบบ Table ที่นำเสนอข้อมูลในรูปแบบตาราง และยังสามารถนำเสนอด้วย Chart ต่างๆ ได้ โดยผู้เขียนได้อธิบายด้วยโปรแกรมในแบบต่างๆ ซึ่งผู้อ่านสามารถนำไปประยุกต์ใช้เพื่อการสร้างรายงานได้ตามต้องการ

## แบบฝึกหัด

1. จงสร้างรายงานเพื่อแสดงผลข้อมูลรหัสนักศึกษา ชื่อนักศึกษา และคะแนนสอบ โดยใช้ข้อมูลจากตาราง Student ในฐานข้อมูลชื่อ mydata.db ในบทที่ 13
2. จงเขียนโปรแกรมสร้างรายงานรูปแบบของกราฟแท่ง เพื่อแสดงผลคะแนนของนักศึกษาแต่ละคน โดยใช้แหล่งข้อมูลเดียวกับข้อ 1.



- บรรทัดที่ 17 เรียกใช้ DataFrame ของคอลัมน์ชื่อวิชาที่เรียนในตัวแปร miss\_data และหาค่าฐานนิยม เก็บไว้ในตัวแปร m
- บรรทัดที่ 18 กำหนดค่า m ให้กับข้อมูลที่สูญหาย ซึ่งก็คือ การแทนที่ข้อมูลนั่นเอง
- บรรทัดที่ 20-21 จัดเก็บข้อมูลใน DataFrame ลงไฟล์ coursedata2.csv

หลังจากได้ไฟล์พร้อมใช้งานแล้ว ผู้เขียนจะได้อธิบายถึงการแสดงผลและนำเสนอข้อมูล (Data Visualization) จากผลการวิเคราะห์ข้อมูล ซึ่งได้จากการเขียนโปรแกรมโดยใช้โมดูล matplotlib ซึ่งผู้อ่านจะได้ศึกษาในหัวข้อถัดไป

## การวิเคราะห์และนำเสนอข้อมูลในรูปแบบของกราฟแผนภูมิด้วยโมดูล matplotlib

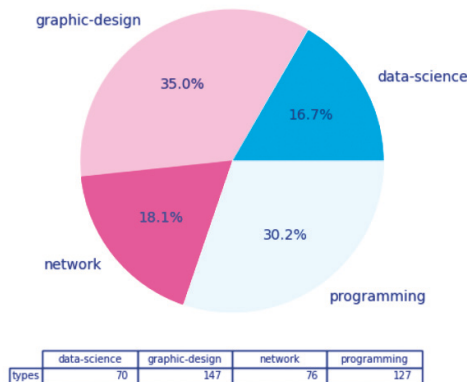
matplotlib เป็นโมดูลหรือไลบรารีที่ใช้สำหรับการนำเสนอข้อมูล ในรูปแบบของกราฟแผนภูมิ สามารถใช้งานร่วมกับข้อมูลในรูปแบบของ DataFrame ที่ได้จากการใช้โมดูล pandas ได้ โดยสามารถนำข้อมูลจาก DataFrame มาจัดทำเป็นแผนภูมิด้วยเมธอด plot ได้ เช่น แผนภูมิวงกลม แผนภูมิเส้น แผนภูมิแท่ง เป็นต้น

### ตัวอย่างการเขียนโปรแกรมวิเคราะห์และนำเสนอข้อมูลด้วยโมดูล matplotlib

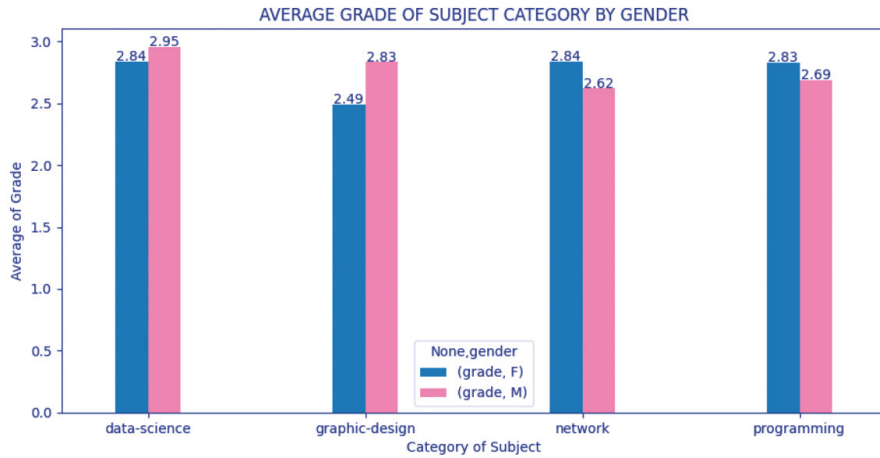
ในส่วนของการวิเคราะห์และนำเสนอข้อมูล ผู้เขียนได้เลือกประเด็นวิเคราะห์จากข้อมูลการลงทะเบียนเรียนรายวิชาเลือกของนักศึกษาใน 3 ประเด็น และแสดงเป็นแผนภูมิดังนี้

1. สัดส่วนเปอร์เซ็นต์ของจำนวนนักศึกษาที่ลงทะเบียนเรียนในแต่ละกลุ่มรายวิชา

PERCENT OF TOTAL STUDENT FOR EACH SUBJECT CATEGORY



## ผลการทำงานของโปรแกรม



จากที่นำเสนอข้อมูลข้างต้น เป็นตัวอย่างของการเขียนโปรแกรมเพื่อค้นหาและค้นพบข้อมูลบางอย่าง ที่จะนำไปสู่การวางแผนจัดการจำนวนหรือสัดส่วนของรายวิชาที่เหมาะสม สำหรับการลงทะเบียนเรียนของนักศึกษาในปีการศึกษาถัดไป

## การจัดการข้อมูลด้วยโมดูล numpy

numpy เป็นโมดูลหรือไลบรารีสำหรับจัดการข้อมูลตัวเลขให้อยู่ในรูปแบบอาร์เรย์ และนำมาใช้เพื่อการคำนวณหรือประมวลผลทางคณิตศาสตร์และสถิติ รวมถึงการนำมาใช้เพื่อนำเสนอข้อมูล ร่วมกับการทำงานของโมดูล matplotlib ซึ่งผู้เขียนจะได้อธิบายถึงขั้นตอนการสร้างอาร์เรย์ และนำข้อมูลมาสร้างเป็นแผนภูมิแบบเรียงซ้อน (Stack Bar Chart)

### การจัดการข้อมูลในรูปแบบอาร์เรย์ด้วยโมดูล numpy

การจัดการข้อมูลตัวเลขให้อยู่ในรูปแบบอาร์เรย์ด้วยโมดูล numpy นั้น สามารถดำเนินการได้ โดยสร้างขึ้นจากข้อมูลแบบเรียงลำดับ (Sequence) ในที่นี้ผู้เขียนจะอธิบายโดยใช้ข้อมูลแบบลิสต์ (List) เมื่อข้อมูลอาร์เรย์ถูกสร้างแล้ว ข้อมูลอาร์เรย์นั้นจะมีเมธอดที่ติดตัวมาด้วย เช่น

- **shape** เป็นเมธอดที่บอกขนาดของอาร์เรย์
- **size** เป็นเมธอดบอกจำนวนสมาชิกของอาร์เรย์
- **ndim** เป็นเมธอดที่บอกจำนวนมิติของอาร์เรย์