

Computer Programming and Developing Applications for Education

Chapter 7: Functions

DTI3302 Computer Programming and Developing Applications for Education

Department of Digital Technology for Education

Faculty of Education, Suan Sunandha Rajabhat University

Content Credit By: Asst.Prof.Nutthapat Kaewrattanapat, PhD.



Pasawut Cheerapakorn

Suan Sunandha Rajabhat University

Course Description:

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน
หลัก การพัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิด
ต่าง ๆ หลักการเบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียน
โปรแกรม ขั้นตอนวิธี การวิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการ
ศึกษา การประเมิน ซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and
development applications, computer programming principles,
computer language, elements of computer language, syntax,
computer programming, algorithms, analysis and design application
for education, software evaluation, candidate teachers able to
developing applications for education.

Algorithm Design

Python Programming

AI for Programming

Course Outline:

- Chapter 1 - Computer Programming
- Chapter 2 - Introduction to Python Programming
- Chapter 3 - Conditional or Decision Statement
- Chapter 4 - Iteration Statement
- Chapter 5 - Strings
- Chapter 6 - Lists, Tuples, Sets, Dictionaries
- **Chapter 7 - Functions**
- Chapter 8 - Object-Oriented Programming: OOP

Measurement and Evaluation:

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test 0%
- การมอบหมายงาน 20%
- สอบ Post-test 15%
- การมีส่วนร่วมในชั้นเรียน 5%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน) 20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ 20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน) 20%

| ร้อยละ | ระดับผลการเรียน | ความหมาย |
|----------|-----------------|---------------------|
| 86 – 100 | A | ดีเยี่ยม |
| 82 – 85 | A- | ดีเยี่ยม |
| 78 – 81 | B+ | ดีมาก |
| 74 – 77 | B | ดี |
| 70 – 73 | B- | ค่อนข้างดี |
| 66 – 69 | C+ | ปานกลางค่อนข้างดี |
| 62 – 65 | C | ปานกลาง |
| 58 – 61 | C- | ปานกลางค่อนข้างอ่อน |
| 54 – 57 | D+ | ค่อนข้างอ่อน |
| 50 – 53 | D | อ่อน |
| 46 – 49 | D- | อ่อนมาก |
| 0 – 45 | F | ตก |

Measurement and Evaluation:

| ครั้งที่ / สัปดาห์ | บทเรียน / หัวข้อ |
|--------------------|-------------------------------------------------------------------------------|
| 1 | แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ (Introduction to Course) |
| 2 | บทที่ 1 การโปรแกรมคอมพิวเตอร์ (Computer Programming) |
| 3 | บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน (Introduction to Python Programming) |
| 4 | บทที่ 3 การโปรแกรมแบบเงื่อนไขหรือตัดสินใจ (Conditional or Decision Statement) |
| 5 | บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration Statement) |
| 6 | บทที่ 5 การโปรแกรมสายอักขระ (String) |
| 7 | บทที่ 6 ลิสต์ ทูเปิล และดิกชันนารี (List, Tuple, and Dictionaries) |
| 8 | บทที่ 7 ฟังก์ชัน (Function) |

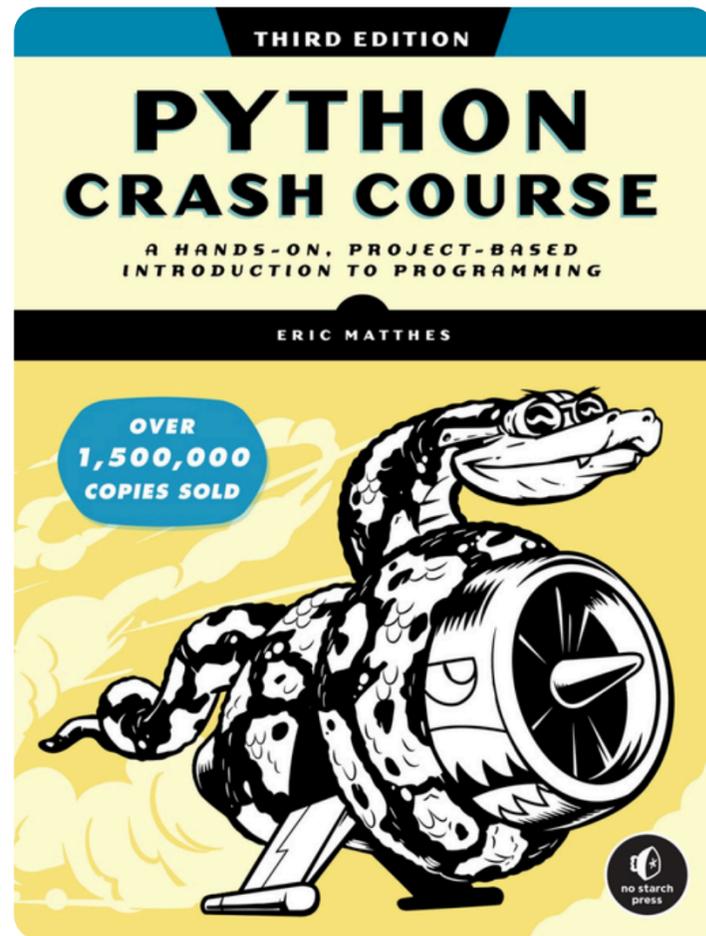
Measurement and Evaluation:

| ครั้งที่ / สัปดาห์ | บทเรียน / หัวข้อ |
|--------------------|----------------------------------------------------------------------|
| 9 | สอบกลางภาค (Midterm Examination) |
| 10 | บทที่ 7 ฟังก์ชัน (Function) (ต่อ) |
| 11 | บทที่ 8 การโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) |
| 12 | บทที่ 8 การโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) (ต่อ) |
| 13 | การเขียนโปรแกรมด้วยปัญญาประดิษฐ์ (AI for Programming) |
| 14 | สอบปลายภาค (Final Examination) |
| 15 | นำเสนอและส่งโครงงาน (Project Pitching and Presentation) |
| 16 | |

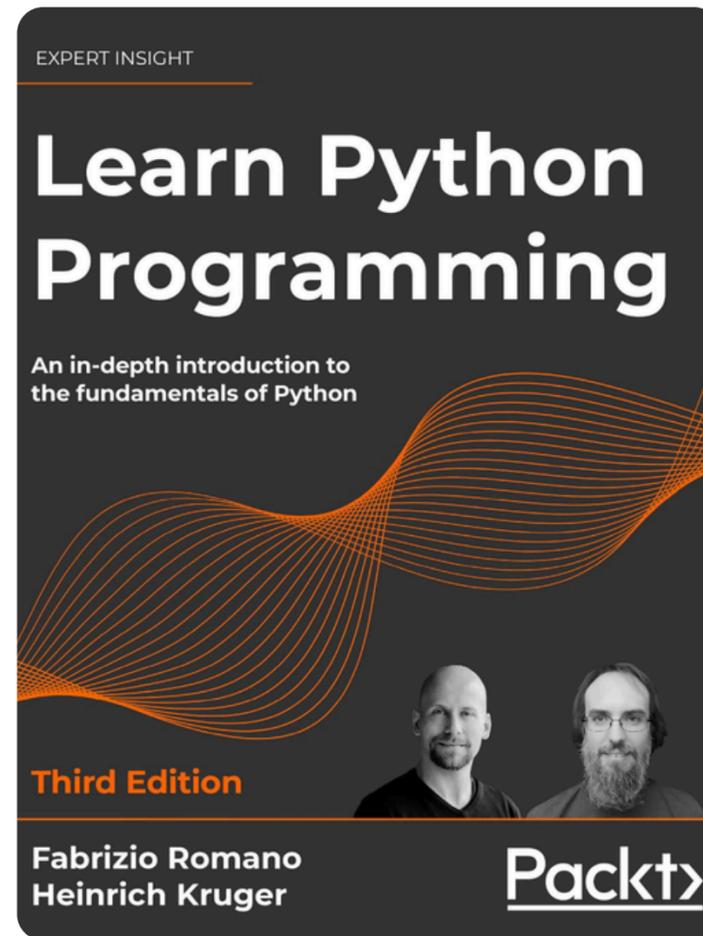
Chapter 7: Functions

1. Parameters and Arguments
2. Void Function
3. Function
4. Return Function
5. Arbitrary Arguments
6. Default Parameter Value
7. Keyword Arguments
8. Passing a List as an Argument

Learning Materials Suggestion:



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Website

- <https://www.w3schools.com/python/>
- <https://realpython.com/>
- <https://docs.python.org/3/tutorial/index.html>
- <https://pythontutor.com/>

Pre Test

Question:

1. ในการเขียนโปรแกรมภาษา Python ฟังก์ชัน (Function) คืออะไร

A

ชุดของคำสั่งที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สี่เหลี่ยม

B

ชุดของคำสั่งและตัวแปรที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สามเหลี่ยม

C

ชุดของตัวแปรที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่วงกลม

D

คำสั่งหรือตัวแปรเดี่ยว ๆ ที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สี่เหลี่ยม

Question:

2. ข้อใด ไม่ใช่ ประโยชน์ของฟังก์ชัน (Function)

A

การใช้ซ้ำของโค้ด
(Code Reusability)

B

การจัดการทรัพยากร
(Resource Management)

C

การปรับแต่งและการเปลี่ยนแปลง
(Customization and Modification)

D

การทำงานอย่างรวดเร็วและแม่นยำ
(Fasten and Accuracy)

Question:

3. ข้อใดกล่าวถูกต้องเกี่ยวกับการทำงานของฟังก์ชัน (Function)

A

เมื่อเรียกใช้งานฟังก์ชัน โปรแกรมจะทำงานตามคำสั่งภายในฟังก์ชันนั้น ๆ

B

เมื่อโปรแกรมทำงานตามคำสั่งภายในฟังก์ชันนั้น ๆ จะสามารถรับค่าอินพุต (input) ได้

C

เมื่อเรียกใช้งานฟังก์ชัน โปรแกรมจะทำงานตามคำสั่งนอกเหนือจากฟังก์ชันนั้น ๆ

D

ส่งค่าเอาต์พุต (output) กลับไปยังโปรแกรมหลัก หรือส่วนอื่น ๆ ของโค้ดได้

Question:

4. การทำงานของโปรแกรมเมื่อเรียกใช้ฟังก์ชัน (Function) เรียกว่าอะไร

A

ค่ารีเทิร์น (Return Value)
และ อ็อกเมนต์เต็ด (Augmented)

B

ค่าพารามิเตอร์ (Parameters)
และ อาร์กิวเมนต์ (Arguments)

C

ค่าพารามิเตอร์ (Parameters)
และ ค่ารีเทิร์น (Return Value)

D

อาร์กิวเมนต์ (Arguments)
และ อ็อกเมนต์เต็ด (Augmented)

Question:

5. ข้อใด ไม่ใช่ ประเภทของฟังก์ชันที่นิยามใหม่ (User-defined Functions)

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

6. จากคำสั่ง เรียกฟังก์ชัน ประเภทนี้ว่าอย่างไร

```
1. def product(price, sale):
2.     total = price - (price*sale/100)
3.     return total
4.
5. p_price = 500
6. p_sale = 50
7. print("สินค้าราคา {p_price} ส่วนลด {p_sale}")
8. print("ราคาที่ต้องชำระ คือ {product(p_price, p_sale)} บาท")
```

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

7. จากคำสั่ง เรียกฟังก์ชัน ประเภทนี้ว่าอย่างไร

```
def triangle_area():  
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))  
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))  
    area = 0.5 * base * height  
    print("พื้นที่ของสามเหลี่ยมคือ: ", area)
```

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

8. จากคำสั่ง เรียกฟังก์ชัน ประเภทนี้ว่าอย่างไร

```
1. def greeting(name) :  
2.     print(f"Hello, {name}")  
3.  
4. greeting("Pasawut")
```

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

9. จากคำสั่ง แสดงผลข้อใด

```
1. def my_function(country = "Thailand"):  
2.     print("I am from " + country)  
3.  
4.  
5. my_function("Sweden")  
6. my_function("India")  
7. my_function()
```

A

I am from Sweden
I am from India
I am from Thailand

B

I am from Sweden
I am from India
I am from Brazil

C

I am from Sweden
I am from Thailand
I am from India

D

I am from Thailand
I am from Thailand
I am from Thailand

Question:

10. จากคำสั่ง แสดงผลข้อใด

```
1. def my_function(food):  
2.     for x in food:  
3.         print(x)  
4.  
5. fruits = ["apple", "banana", "cherry"]  
6. my_function(fruits)
```

A

banana
apple
cherry

B

cherry
banana
apple

C

apple
banana
cherry

D

food
x
fruits

ฟังก์ชัน (Functions)

ฟังก์ชัน (Functions) ในภาษา Python คือ ชุดของคำสั่งที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สี่เหลี่ยม

เมื่อเรียกใช้งานฟังก์ชัน โปรแกรมจะทำงานตามคำสั่งภายในฟังก์ชันนั้น ๆ และสามารถรับค่าอินพุต (input) และส่งค่าเอาต์พุต (output) กลับไปยังโปรแกรมหลักหรือส่วนอื่น ๆ ของโค้ดได้ ซึ่งจะเรียกว่า ค่าพารามิเตอร์ (Parameters) และ อาร์กิวเมนต์ (Arguments).

ประโยชน์ของฟังก์ชัน

1. การใช้ซ้ำของโค้ด (Code Reusability)
2. การจัดการโค้ดและอ่านโค้ดได้ดีขึ้น
3. การทดสอบและการดูแลรักษา (Testing and Maintenance)
4. การจัดการทรัพยากร (Resource Management)
5. การออกแบบ (Design)
6. การปรับแต่งและการเปลี่ยนแปลง (Customization and Modification)

การเปรียบเทียบการเขียนโปรแกรมแบบ Procedural และ Function

การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming)

```
1. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
2. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
3. area = length * width
4. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

5. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
6. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
7. area = length * width
8. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

9. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
10. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
11. area = length * width
12. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")
```

การเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming)

```
1. def rectangle():
2.     length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
3.     width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
4.     area = length * width
5.     print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

6. rectangle()

7. rectangle()

8. rectangle()
```

การเปรียบเทียบการเขียนโปรแกรมแบบ Procedural และ Function

การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming)

```
1. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
2. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
3. area = length * width
4. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

5. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
6. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
7. area = length * width
8. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

9. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
10. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
11. area = length * width
12. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")
```

การเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming)

```
1. def rectangle():
2.     length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
3.     width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
4.     area = length * width
5.     print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

6. rectangle()

7. rectangle()

8. rectangle()
```

Function

Calling a Function

ประเภทของ Function

1. ฟังก์ชันพื้นฐานทั่วไป (Built-in Function)

เช่น print() len() input()

2. ฟังก์ชันที่นิยามใหม่ (User-defined Functions)

- ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)
- ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)
- ฟังก์ชันแบบคืนค่า (Return Function)

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันสำหรับแสดงเมนูการทำงาน และรับข้อมูลเมนู ดังนี้

เมนูฟังก์ชันสำหรับการหาพื้นที่

กด 1 หาพื้นที่สี่เหลี่ยม (Rectangle Area)

กด 2 หาพื้นที่สามเหลี่ยม (Triangle Area)

กด 3 หาพื้นที่วงกลม (Circle Area)

```
def show_menu():  
    print("เมนูการทำงาน: ")  
    print("กด 1: หาพื้นที่สี่เหลี่ยม (Rectangle Area)")  
    print("กด 2: หาพื้นที่สามเหลี่ยม (Triangle Area)")  
    print("กด 3: หาพื้นที่วงกลม (Circle Area)")  
    choice = input("โปรดเลือกหมายเลขเมนู: ")  
    if choice == '1':  
        print("คุณเลือกหาพื้นที่สี่เหลี่ยม")  
    elif choice == '2':  
        print("คุณเลือกหาพื้นที่สามเหลี่ยม")  
    elif choice == '3':  
        print("คุณเลือกหาพื้นที่วงกลม")  
    else:  
        print("โปรดเลือกหมายเลขเมนูที่ถูกต้อง")
```

```
show_menu()
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันหาพื้นที่สี่เหลี่ยม (Rectangle Area)

```
def rectangle_area():  
    length = float(input("ป้อนความยาวของสี่เหลี่ยม: "))  
    width = float(input("ป้อนความกว้างของสี่เหลี่ยม: "))  
    area = length * width  
    print("พื้นที่ของสี่เหลี่ยมคือ: ", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันหาพื้นที่สามเหลี่ยม (Triangle Area)

```
def triangle_area():  
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))  
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))  
    area = 0.5 * base * height  
    print("พื้นที่ของสามเหลี่ยมคือ: ", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันหาพื้นที่วงกลม (Circle Area)

```
import math

def circle_area():
    radius = float(input("ป้อนรัศมีของวงกลม: "))
    area = math.pi * radius**2
    print("พื้นที่ของวงกลมคือ: ", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

```
import math
def show_menu():
    print("เมนูการทำงาน:")
    print("กด 1: หาพื้นที่สี่เหลี่ยม (Rectangle Area)")
    print("กด 2: หาพื้นที่สามเหลี่ยม (Triangle Area)")
    print("กด 3: หาพื้นที่วงกลม (Circle Area)")
    choice = input("โปรดเลือกหมายเลขเมนู: ")
    if choice == '1':
        print("คุณเลือกหาพื้นที่สี่เหลี่ยม")
    elif choice == '2':
        print("คุณเลือกหาพื้นที่สามเหลี่ยม")
    elif choice == '3':
        print("คุณเลือกหาพื้นที่วงกลม")
    else:
        print("โปรดเลือกหมายเลขเมนูที่ถูกต้อง")
```

```
def rectangle_area():
    length = float(input("ป้อนความยาวของสี่เหลี่ยม: "))
    width = float(input("ป้อนความกว้างของสี่เหลี่ยม: "))
    area = length * width
    print("พื้นที่ของสี่เหลี่ยมคือ: ", area)

def triangle_area():
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))
    area = 0.5 * base * height
    print("พื้นที่ของสามเหลี่ยมคือ: ", area)

def circle_area():
    radius = float(input("ป้อนรัศมีของวงกลม: "))
    area = math.pi * radius**2
    print("พื้นที่ของวงกลมคือ: ", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

```
import math
def show_menu():
    print("เมนูการทำงาน:")
    print("กด 1: หาพื้นที่สี่เหลี่ยม (Rectangle Area)")
    print("กด 2: หาพื้นที่สามเหลี่ยม (Triangle Area)")
    print("กด 3: หาพื้นที่วงกลม (Circle Area)")
    choice = input("โปรดเลือกหมายเลขเมนู: ")
    if choice == '1':
        rectangle_area()
    elif choice == '2':
        triangle_area()
    elif choice == '3':
        circle_area()
    else:
        print("โปรดเลือกหมายเลขเมนูที่ถูกต้อง")
```

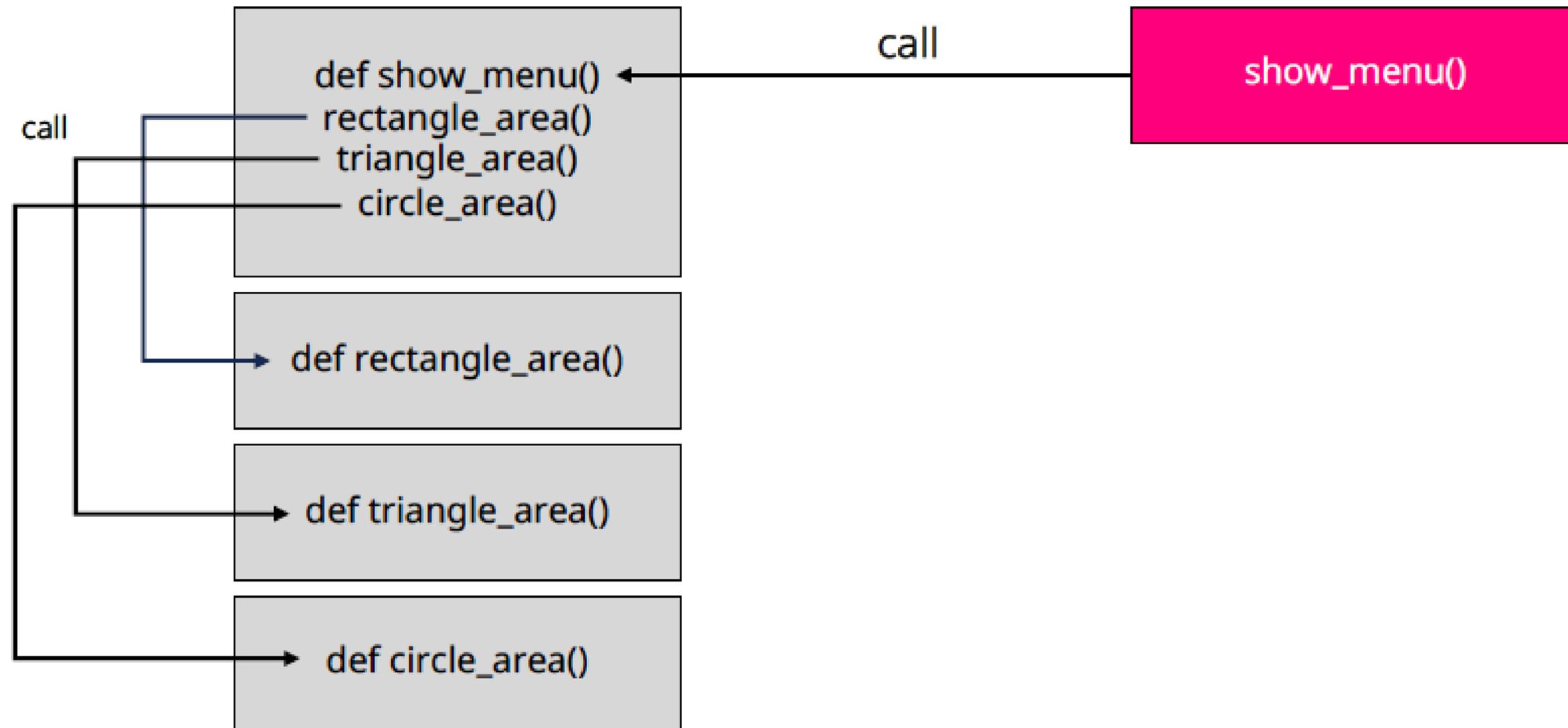
```
def rectangle_area():
    length = float(input("ป้อนความยาวของสี่เหลี่ยม: "))
    width = float(input("ป้อนความกว้างของสี่เหลี่ยม: "))
    area = length * width
    print("พื้นที่ของสี่เหลี่ยมคือ: ", area)

def triangle_area():
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))
    area = 0.5 * base * height
    print("พื้นที่ของสามเหลี่ยมคือ: ", area)

def circle_area():
    radius = float(input("ป้อนรัศมีของวงกลม: "))
    area = math.pi * radius**2
    print("พื้นที่ของวงกลมคือ: ", area)
```

```
show_menu()
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)



ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

Parameter คือ ตัวแปรที่รับค่าจากภายนอกเมื่อมีการเรียกฟังก์ชัน

Parameter

```
1. def greeting(name):  
2.     print(f"Hello, {name}")  
3.  
4.  
5. greeting("Pasawut")
```

Argument

Argument คือ ค่าที่ส่งเข้าไปให้ฟังก์ชัน

```
▶ def greeting(name):  
    print(f"Hello, {name}")  
  
greeting("Pasawut")  
... Hello, Pasawut
```

```
▶ def greeting(name):  
    print(f"Hello, {name}")  
  
greeting("Pasawut")  
greeting("Nutthapat")  
... Hello, Pasawut  
    Hello, Nutthapat
```

ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

สถานการณ์: ต้องการสร้าง function สำหรับทักทาย โดยมีการรับค่า parameter เพศ (gender) และชื่อ (name)

ถ้า gender เป็น male แสดง “Hello, Mr.Name”

ถ้า gender เป็น female แสดง “Hello, Ms.Name”

ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

สถานการณ์: ต้องการสร้าง function สำหรับแปลงอุณหภูมิโดยมีการส่งค่า Arguments และแสดงผลการแปลงอุณหภูมิ

- show_menu

โปรดเลือกการแปลง:

1. Fahrenheit เป็น Celsius
2. Celsius เป็น Fahrenheit

- fahrenheit_to_celsius สูตร celsius = (fahrenheit - 32) * 5/9
- celsius_to_Fahrenheit สูตร fahrenheit = (celsius * 9/5) + 32

ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

สถานการณ์: ต้องการสร้าง function สำหรับแปลงอุณหภูมิโดยมีการส่งค่า Arguments และแสดงผลการแปลงอุณหภูมิ

- show_menu

โปรดเลือกการแปลง:

1. Fahrenheit เป็น Celsius
2. Celsius เป็น Fahrenheit

```
โปรดเลือกการแปลง:  
1. Fahrenheit เป็น Celsius  
2. Celsius เป็น Fahrenheit  
1  
ป้อนอุณหภูมิในฟาเรนไฮต์: 44  
44.0 องศาฟาเรนไฮต์ เท่ากับ 6.67 องศาเซลเซียส
```

- fahrenheit_to_celsius สูตร celsius = (fahrenheit - 32) * 5/9
- celsius_to_Fahrenheit สูตร fahrenheit = (celsius * 9/5) + 32

ฟังก์ชันแบบคืนค่า (Return Function)

```
1. def add(num1, num2):  
2.     result = num1 + num2  
3.     return result  
4.  
5. print(add(5, 6))
```

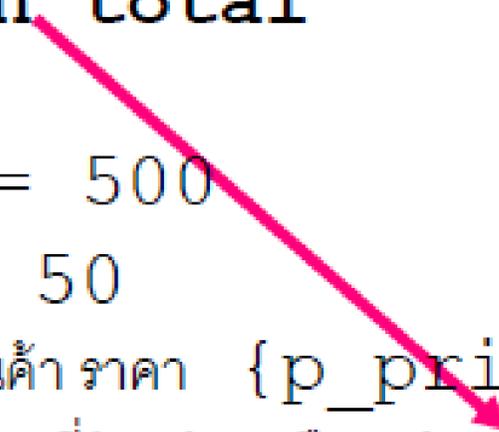
Parameter

Return

Argument

ฟังก์ชันแบบคืนค่า (Return Function)

```
1. def product(price, sale):  
2.     total = price - (price*sale/100)  
3.     return total  
4.  
5. p_price = 500  
6. p_sale = 50  
7. print("สินค้าราคา {p_price} ส่วนลด {p_sale}")  
8. print("ราคาที่ต้องชำระ คือ {product(p_price, p_sale)} บาท")
```



ฟังก์ชันแบบคืนค่า (Return Function)

```
1. def fahrenheit_to_celsius(fahrenheit):
2.     celsius = (fahrenheit - 32) * 5/9
3.     return celsius
4.
5. def celsius_to_fahrenheit(celsius):
6.     fahrenheit = (celsius * 9/5) + 32
7.     return fahrenheit
8.
9. def show_menu():
10.    choice = input("โปรดเลือกการแปลง:\n1. Fahrenheit เป็น Celsius\n2. Celsius เป็น Fahrenheit\n")
11.    if choice == '1':
12.        fahrenheit = float(input("ป้อนอุณหภูมิในฟาเรนไฮต์: "))
13.        celsius = fahrenheit_to_celsius(fahrenheit)
14.        print(f"{fahrenheit} องศาฟาเรนไฮต์เท่ากับ {celsius:.2f} องศาเซลเซียส")
15.    elif choice == '2':
16.        celsius = float(input("ป้อนอุณหภูมิในเซลเซียส: "))
17.        fahrenheit = celsius_to_fahrenheit(celsius)
18.        print(f"{celsius} องศาเซลเซียสเท่ากับ {fahrenheit:.2f} องศาฟาเรนไฮต์" )
19.    else:
20.        print("โปรดเลือกตัวเลือกที่ถูกต้อง")
21.
22. show_menu()
```

ฟังก์ชันแบบมีตัวแปรอาร์กิวเมนต์ที่ไม่แน่นอน (Arbitrary Arguments)

```
1. def my_function(*args):
2.     for text in args:
3.         print("Country:", text)
4.
5. my_function("Thailand", "Japan", "China")
6. print("*"*50)
7. my_function("Thailand", "Japan", "China", "South Korea", "USA")
```

```
1 def my_function(*args):
2     for arg in args:
3         print("Country:",arg)
4
5 my_function("Thailand", "Japan", "China")
6 print("*"*50)
7 my_function("Thailand", "Japan", "China", "South Korea", "USA")
```

```
Country: Thailand
Country: Japan
Country: China
*****
Country: Thailand
Country: Japan
Country: China
Country: South Korea
Country: USA
```

ฟังก์ชันแบบระบุค่าเฉพาะของอาร์กิวเมนต์ (Keyword Arguments)

สถานการณ์: ต้องการสร้าง function สำหรับทักทาย โดยมีการรับค่า parameter เพศ (gender) และชื่อ (name)

ถ้า gender เป็น male แสดง “Hello, Mr.Name”

ถ้า gender เป็น female แสดง “Hello, Ms.Name”

ฟังก์ชันแบบระบุค่าเฉพาะของอาร์กิวเมนต์ (Keyword Arguments)

```
1. def greet(name, gender):
2.     if gender == "male":
3.         print(f"Hello, Mr.{name}")
4.     elif gender == "female":
5.         print(f"Hello, Ms.{name}")
6.
7. greet(gender="male", name="Nutthapat")
8. greet(name="Wichuda", gender="female")
```

ฟังก์ชันแบบระบุค่าเฉพาะของอาร์กิวเมนต์และมีค่าอาร์กิวเมนต์ไม่แน่นอน (Arbitrary Keyword Arguments, **kwargs)

1. `def my_function(**kwargs):`
2. `for key, value in kwargs.items():`
3. `print(f"{key}: {value}")`
4. `print("-"*50)`
5. `my_function(city="Thailand", region="Asia")`
6. `my_function(city="Japan", region="Asia", latitude="36°00' N", longitude="138°00' E")`

```
1 def my_function(**kwargs):
2     for key, value in kwargs.items():
3         print(f"{key}: {value}")
4     print("-"*50)
5
6 my_function(city="Thailand", region="Asia")
7 my_function(city="Japan", region="Asia", latitude="36° 00' N", longitude="138° 00' E")
8
```

```
city: Thailand
region: Asia
```

```
-----
city: Japan
region: Asia
latitude: 36° 00' N
longitude: 138° 00' E
-----
```

ฟังก์ชันแบบระบุค่าโดยปริยายให้กับพารามิเตอร์ (Default Parameter Value)

```
1. def my_function(country = "Thailand") :  
2.     print("I am from " + country)  
3.  
4.  
5. my_function("Sweden")  
6. my_function("India")  
7. my_function()  
8. my_function("Brazil")  
9. my_function()
```

```
1 def my_function(country = "Thailand") :  
2     print("I am from " + country)  
3  
4 my_function("Sweden")  
5 my_function("India")  
6 my_function()  
7 my_function("Brazil")  
8 my_function()
```

```
I am from Sweden  
I am from India  
I am from Thailand  
I am from Brazil  
I am from Thailand
```

ฟังก์ชันแบบส่งผ่านค่าประเภทลิสต์อาร์กิวเมนต์ (Passing a List as an Argument)

```
1. def my_function(food):  
2.     for x in food:  
3.         print(x)  
4.  
5. fruits = ["apple", "banana", "cherry"]  
6. my_function(fruits)
```

```
1 def my_function(food):  
2     for x in food:  
3         print(x)  
4  
5 fruits = ["apple", "banana", "cherry"]  
6 my_function(fruits)
```

```
apple  
banana  
cherry
```

ฟังก์ชันแบบส่งผ่านค่าประเภทลิสต์อาร์กิวเมนต์ (Passing a List as an Argument)

สถานการณ์: ต้องการสร้าง function สำหรับรับค่าคะแนนนักเรียนแบบ list และคำนวณหาค่าเฉลี่ย

```
1. def _____(_____):
2.     total = _____(scores)
3.     average = total / _____(scores)
4.     return average
5.
6. student_scores = [85, 90, 92, 88, 95]
7. _____= calculate_average(_____)
8. print("คะแนนเฉลี่ยของนักเรียน: ", average_score)
```

Assignment 6 - มอบหมายงานครั้งที่ 6

1. ให้นักศึกษาเขียนฟังก์ชัน (Function) **รับค่าสำหรับการแปลงหน่วยน้ำหนักระหว่าง กรัม (g.) และ กิโลกรัม (kg.)** โดยสามารถเขียนฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function) หรือ ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function) หรือ ฟังก์ชันแบบคืนค่า (Return Function) ก็ได้

```
เมนูแปลงหน่วยน้ำหนัก
1. กรัม -> กิโลกรัม
2. กิโลกรัม -> กรัม
เลือกเมนู: 1
กรอกน้ำหนัก (กรัม): 10000
10000.0 g = 10.0 kg
```

```
เมนูแปลงหน่วยน้ำหนัก
1. กรัม -> กิโลกรัม
2. กิโลกรัม -> กรัม
เลือกเมนู: 2
กรอกน้ำหนัก (กิโลกรัม): 10
10.0 kg = 10000.0 g
```

สูตร: กรัม = กิโลกรัม * 1000 และ กิโลกรัม = กรัม / 1000

Assignment 6 - มอบหมายงานครั้งที่ 6

2. ให้นักศึกษาเขียนฟังก์ชัน (Function) **รับค่าสำหรับการแปลงหน่วยความยาวระหว่าง ไมล์ (mi.) และ กิโลเมตร (km.)** โดยสามารถเขียนฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function) หรือ ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function) หรือ ฟังก์ชันแบบคืนค่า (Return Function) ก็ได้

```
เมนูแปลงหน่วยความยาว
1. ไมล์ -> กิโลเมตร
2. กิโลเมตร -> ไมล์
เลือกเมนู: 1
กรอกระยะทาง (ไมล์): 100
100.0 mi = 160.93 km
```

```
เมนูแปลงหน่วยความยาว
1. ไมล์ -> กิโลเมตร
2. กิโลเมตร -> ไมล์
เลือกเมนู: 2
กรอกระยะทาง (กิโลเมตร): 160.93
160.93 km = 100.00 mi
```

สูตร: กิโลเมตร = ไมล์ * 1.60934 และ ไมล์ = กิโลเมตร / 1.60934

Post Test

Question:

1. ในการเขียนโปรแกรมภาษา Python ฟังก์ชัน (Function) คืออะไร

A

ชุดของคำสั่งที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สี่เหลี่ยม

B

ชุดของคำสั่งและตัวแปรที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สามเหลี่ยม

C

ชุดของตัวแปรที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่วงกลม

D

คำสั่งหรือตัวแปรเดี่ยว ๆ ที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สี่เหลี่ยม

Question:

2. ข้อใด ไม่ใช่ ประโยชน์ของฟังก์ชัน (Function)

A

การใช้ซ้ำของโค้ด
(Code Reusability)

B

การจัดการทรัพยากร
(Resource Management)

C

การปรับแต่งและการเปลี่ยนแปลง
(Customization and Modification)

D

การทำงานอย่างรวดเร็วและแม่นยำ
(Fasten and Accuracy)

Question:

3. ข้อใดกล่าวถูกต้องเกี่ยวกับการทำงานของฟังก์ชัน (Function)

A

เมื่อเรียกใช้งานฟังก์ชัน โปรแกรมจะทำงานตาม คำสั่งภายในฟังก์ชันนั้น ๆ

B

เมื่อโปรแกรมทำงานตาม คำสั่งภายในฟังก์ชันนั้น ๆ จะ สามารถรับค่าอินพุต (input) ได้

C

เมื่อเรียกใช้งานฟังก์ชัน โปรแกรมจะทำงานตามคำสั่ง นอกเหนือจากฟังก์ชันนั้น ๆ

D

ส่งค่าเอาต์พุต (output) กลับไปยังโปรแกรมหลัก หรือส่วนอื่น ๆ ของโค้ดได้

Question:

4. การทำงานของโปรแกรมเมื่อเรียกใช้ฟังก์ชัน (Function) เรียกว่าอะไร

A

ค่ารีเทิร์น (Return Value)
และ อ็อกเมนต์เต็ด (Augmented)

B

ค่าพารามิเตอร์ (Parameters)
และ อาร์กิวเมนต์ (Arguments)

C

ค่าพารามิเตอร์ (Parameters)
และ ค่ารีเทิร์น (Return Value)

D

อาร์กิวเมนต์ (Arguments)
และ อ็อกเมนต์เต็ด (Augmented)

Question:

5. ข้อใด ไม่ใช่ ประเภทของฟังก์ชันที่นิยามใหม่ (User-defined Functions)

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

6. จากคำสั่ง เรียกฟังก์ชัน ประเภทนี้ว่าอย่างไร

```
1. def product(price, sale):
2.     total = price - (price*sale/100)
3.     return total
4.
5. p_price = 500
6. p_sale = 50
7. print("สินค้าราคา {p_price} ส่วนลด {p_sale}")
8. print("ราคาที่ต้องชำระ คือ {product(p_price, p_sale)} บาท")
```

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

7. จากคำสั่ง เรียกฟังก์ชัน ประเภทนี้ว่าอย่างไร

```
def triangle_area():  
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))  
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))  
    area = 0.5 * base * height  
    print("พื้นที่ของสามเหลี่ยมคือ: ", area)
```

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

8. จากคำสั่ง เรียกฟังก์ชัน ประเภทนี้ว่าอย่างไร

```
1. def greeting(name) :  
2.     print(f"Hello, {name}")  
3.  
4. greeting("Pasawut")
```

A

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์
(Void Function)

C

ฟังก์ชันแบบคืนค่า
(Return Function)

B

ฟังก์ชันแบบมีค่าพารามิเตอร์
(Function)

D

ฟังก์ชันแบบไม่คืนค่า
(No-Return Function)

Question:

9. จากคำสั่ง แสดงผลข้อใด

```
1. def my_function(country = "Thailand"):  
2.     print("I am from " + country)  
3.  
4.  
5. my_function("Sweden")  
6. my_function("India")  
7. my_function()
```

A

I am from Sweden
I am from India
I am from Thailand

B

I am from Sweden
I am from India
I am from Brazil

C

I am from Sweden
I am from Thailand
I am from India

D

I am from Thailand
I am from Thailand
I am from Thailand

Question:

10. จากคำสั่ง แสดงผลข้อใด

```
1. def my_function(food):  
2.     for x in food:  
3.         print(x)  
4.  
5. fruits = ["apple", "banana", "cherry"]  
6. my_function(fruits)
```

A

banana
apple
cherry

B

cherry
banana
apple

C

apple
banana
cherry

D

food
x
fruits

Computer Programming and
Developing Applications for Education

Thank You

DTI3302 Computer Programming and Developing Applications for Education

Department of Digital Technology for Education

Faculty of Education, Suan Sunandha Rajabhat University



Pasawut Cheerapakorn

Suan Sunandha Rajabhat University