ชื่อ_____รหัสนิสิต_____หมู่_____

Lab 9 - เมธอด ภาค 1

ในภาษาไพทอนเราสามารถแบ่งการทำงานของโปรแกรมเป็นส่วนย่อย ๆ ได้โดยใช้ฟังก์ชัน ในภาษา C# เราก็สามารถแบ่งการทำงานของโปรแกรมใด ๆ เป็นส่วนย่อย ๆ ได้เช่นเดียวกันโดยใช้ "**เมธอด**"

1. การประกาศเมธอด

เมธอดในภาษา C# นั้นจำเป็นต้องประกาศไว้ก่อนที่จะมีการเรียกใช้เสมอ (เช่นเดียวกันกับฟังก์ชันในภาษาไพทอนที่จำเป็นต้องนิยามไว้ก่อนเช่นกัน) โดยประกาศไว้นอกเมธอดใด ๆ แต่อยู่ใน Class เดียวกัน มีรูปแบบของการประกาศเมธอดดังนี้

```
static  <ชนิดของข้อมูลที่เมธอดจะคืนค่า>  <ชื่อเมธอด>  ( <รายการพารามิเตอร์> )
{
    // โปรแกรมของเมธอด
}
```

ตัวอย่างของการประกาศเมธอด Cal_Area() ซึ่งเป็นเมธอดสำหรับคำนวณหาพื้นที่ของรูปสามเหลี่ยมเมื่อทราบความยาวฐานและความสูง เป็นดังต่อไปนี้

	การประกาศเมธอด Cal_Area() ในภาษา C#	การนิยามฟังก์ชัน Cal_Area() ในภาษาไพทอน
1	static double Cal_Area(double base, double height)	def Cal_Area (base, height):
2	{	area = 0.5*base*height
3	double area;	return area
4	area = 0.5*base*height;	
5	return area;	
6	}	

จากการประกาศเมธอด Cal_Area() ในภาษา C# ข้างต้น เนื่องจากเราต้องการคำนวณหาพื้นที่ที่คำตอบเป็นจำนวนจริงใด ๆ จึงต้องเขียน double ไว้หลังคำว่า static แต่ก่อนชื่อเมธอด Cal_Area ในบรรทัดที่ 1

รายการพารามิเตอร์ คือรายการของตัวแปรที่เราประกาศไว้เพื่อให้ติดต่อการส่วนของโปรแกรมหลัก โดยในโปรแกรมหลักสามารถส่งค่าต่าง ๆ เข้ามาเก็บในตัวแปรที่เป็นพารามิเตอร์นี้ได้เวลาเรียกได้ อาทิเช่น หากเราต้องการคำนวณหาพื้นที่รูปสามเหลี่ยมที่มีความยาวฐานเท่ากับ 5 และความสูงเท่ากับ 8 เราสามารถเรียกใช้เมธอดได้ดังนี้

```
Cal_Area(5, 8)
```

หลังจากเรียกใช้เมธอดข้างต้น ค่าของ 5 และ 8 จะถูก copy ส่งไปเก็บไว้ในตัวแปร base และ height ตามลำดับ ทำให้ตัวแปร area สามารถนำค่าเหล่านั้นมาคำนวณได้ในบรรทัดที่ 4

เมื่อคำนวณได้คำตอบแล้ว การส่งค่ากลับไปยังโปรแกรมหลักสามารถทำได้โดยใช้คำสั่ง return เช่นเดียวกันกับคำสั่ง return ในภาษาไพทอน

2. การคืนค่าของเมธอด

การคืนค่าของเมธอดคือการส่งค่ากลับไปยังโปรแกรมหลัก โดยจะค่าที่คืนจะถูกนำไปวางไว้ ณ ตำแหน่งที่เรียกใช้เมธอด ดังตัวอย่างการเรียกใช้เมธอด Cal_Area (ที่ได้ประกาศไว้แล้วในหัวข้อที่ 1) ในตัวอย่างที่ 2.1 ข้างล่างนี้ เมธอดจะคืนค่าที่คำนวณได้กลับมาวางไว้ ณ ตำแหน่งที่เรียก ดังตัวอย่างที่ 2.2

ตัวอย่างที่ 2.1	ตัวอย่างที่ 2.2
double area; area = **Cal_Area(10, 12)**;	double area; area = **60**;

ทั้งนี้เมธอดบางเมธอดอาจไม่มีการคืนค่าก็ได้ (เช่นเดียวกันกับฟังก์ชันในภาษาไพทอนที่อาจไม่คืนค่าก็ได้เช่นกัน) โดยหากไม่มีการคืนค่าให้ใส่คำว่า void ไว้ในพื้นที่ของการประกาศชนิดของข้อมูลที่จะคืนค่า (หลังคำว่า static ก่อนชื่อของเมธอด) ตัวอย่างเช่นเมธอด Show_Area() ที่ใช้ในการแสดงข้อมูลของพื้นที่สามเหลี่ยม (เป็นเมธอดใช้แสดงข้อมูลออกทางหน้าจอเท่านั้น) สามารถเขียนได้ดังตัวอย่างที่ 2.3 ซึ่งผลลัพธ์จะมีค่าเทียบเหมือนกับตัวอย่างที่ 2.4

ตัวอย่างที่ 2.3	ตัวอย่างที่ 2.4

static void ShowArea(double n)
{
Console.WriteLine("Area of Triangle is {0}",n);
}
 	
static void Main(){
area = 15;
ShowArea(area);
}
	static void Main(){
	area = 15;
	Console.WriteLine("Area of Triangle is {0}",area);
	}

ให้นักเรียนลองฝึกทำความเข้าใจโดยทำแบบฝึกหัดที่ 1.1 และ 1.2 ต่อไปนี้ ซึ่งเป็นการเขียนเมธอดที่ให้ผลลัพธ์ของการทำงานเหมือนกัน แต่เมธอดทั้งสองข้อเขียนไม่เหมือนกัน (คืนค่ากับไม่คืนค่า)

แบบฝึกหัดที่ 1.1 : การเรียกเมธอด: คำนวณพื้นที่ 1

ข้อนี้เราจะพิจารณาการเรียกใช้งานเมธอด มีเมธอดสองเมธอดคือ CalSquare และ CalRectangle.

- เมธอด CalSquare คำนวณพื้นที่ของสี่เหลี่ยมจตุรัส เมื่อได้รับความยาวด้าน
- เมธอด CalRectangle คำนวณพื้นที่ของสี่เหลี่ยม เมื่อได้รับความกว้างและความยาว

งานของคุณคือเติมโปรแกรมด้านล่างให้สมบูรณ์

ตัวอย่างผลลัพธ์ของโปรแกรมในแบบฝึกหัดที่ 1.1

```
Calculate the area of a rectangle.
Height = 1
Width = 2
The area of a rectangle = 2
Calculate the area of a square.
Length of side = 3
The area of a square = 9
```

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)

```
using System;
namespace simplemethod
{
    class Program
    {
        static void CalSquare(int x) {
            int area;
            area = x*x;
            Console.WriteLine("The area of a square = {0}",area);
        }

        static void CalRectangle(int h, int w) {
            double area;
            area = h*w;
            Console.WriteLine("The area of a rectangle = {0}",area);
        }

        public static void Main(string[] args)
        {
            Console.WriteLine("Calculate the area of a rectangle.");
            Console.Write("Height = ");
            int a = int.Parse(Console.ReadLine());
            Console.Write("Width = ");
            int b = int.Parse(Console.ReadLine());

            ____________________________________

            Console.WriteLine("Calculate the area of a square.");
            Console.Write("Length of side = ");
            int c = int.Parse(Console.ReadLine());
            ____________________________________
        }
    }
}
```

แบบฝึกหัดที่ 1.2 : การเรียกเมธอด: คำนวณพื้นที่ 2

ข้อนี้เราจะพิจารณาการเรียกใช้งานเมธอด มีเมธอดสองเมธอดคือ CalSquare และ CalRectangle.

- เมธอด CalSquare คำนวณพื้นที่ของสี่เหลี่ยมจตุรัส เมื่อได้รับความยาวด้าน
- เมธอด CalRectangle คำนวณพื้นที่ของสี่เหลี่ยม เมื่อได้รับความกว้างและความยาว

งานของคุณคือเติมโปรแกรมด้านล่างให้สมบูรณ์

ตัวอย่างผลลัพธ์ของโปรแกรมในแบบฝึกหัดที่ 1.2

```
Calculate the area of a rectangle.
Height = 1
Width = 2
The area of a rectangle = 2
Calculate the area of a square.
Length of side = 3
The area of a square = 9
```

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)

```
using System;
namespace simplemethod
{
    class Program
    {
        static double cal_square(int x) {
            double area;
            area = x*x;
            return area;
        }

        static double cal_rectangle(int h, int w) {
            return h*w;
        }

        public static void Main(string[] args)
        {
            Console.WriteLine("Calculate the area of a rectangle.");
            Console.Write("Height = ");
            int a = int.Parse(Console.ReadLine());
            Console.Write("Width = ");
            int b = int.Parse(Console.ReadLine());
            Console.WriteLine("The area of a rectangle = {0}",_________________);

            Console.WriteLine("Calculate the area of a square.");
            Console.Write("Length of side = ");
            int c = int.Parse(Console.ReadLine());
            double area;
            area = _________________
            Console.WriteLine("The area of a square = {0}",area);
            Console.ReadKey();
        }
    }
}
```

3. ฝึกเขียนเมธอด

แบบฝึกหัดที่ 3.1 : พื้นที่วงกลม

โปรแกรมต่อไปนี้ทำงานคล้ายกับโปรแกรมในตัวอย่างที่ผ่านมา แต่โปรแกรมจะทำการคำนวณพื้นที่ของวงกลมแทนที่จะเป็นสามเหลี่ยม โดยรับค่ารัศมีของวงกลมจากผู้ใช้ แต่โปรแกรมยังขาดส่วนสำคัญที่เว้นว่างเอาไว้คือเมธอด CircleArea ที่จะทำการคำนวณค่าพื้นที่จากรัศมี

จงเขียนเมธอดดังกล่าวให้สมบูรณ์

ตัวอย่างผลลัพธ์ของโปรแกรมในแบบฝึกหัดที่ 1.3
Enter radius:**15** The area of the circle is 706.86.

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)

```
using System;
class Circle {

   ____________________________________________
   ____________________________________________
   ____________________________________________
   ____________________________________________
   ____________________________________________

   static void Main(){
     Console.Write("Enter radius:");
     double radius = double.Parse(Console.ReadLine());
     Console.WriteLine("The area of the circle is {0:f2}.",
                       CircleArea(radius));
   }
}
```

แบบฝึกหัดที่ 3.2 : แก้โปรแกรมพิมพ์ดาว

พิจารณาโปรแกรมต่อไปนี้

```
using System;
class MyClass {
  static void PrintLine()
  {
     int i = 0;
     while(i < 20)
     {
        Console.Write('*');
        i++;
     }
     Console.WriteLine();
  }

  static void Main() {
     PrintLine();
     PrintLine();
  }
}
```

จงดัดแปลงโปรแกรมข้างต้นให้สามารถทำงานดังต่อไปนี้

แบบฝึกหัดที่ 3.2.1 : เมธอด PrintLongerLine

ประกาศเมธอดใหม่ชื่อว่า PrintLongerLine ที่ทำงานเหมือนกับ PrintLine แต่ต่างกันตรงที่พิมเมเธอด PrintLongerLine จะพิมพ์ จำนวน 29 ดวงแทนที่จะเป็น 20 ดวง

เมธอด PrintLongerLine ที่เขียนได้คือ

แบบฝึกหัดที่ 3.2.2 : เมธอด Main

ให้นักเรียนแก้ไขการเรียกใช้คำสั่งในเมธอด Main เพื่อให้แสดงผลลัพธ์ดังต่อไปนี้

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

*หมายเหตุ: สามารถเรียกใช้เมธอด PrintLine ได้ด้วย

เมธอด Main ที่เขียนได้คือ

`static void Main()`
`{`
 	
`}`

แบบฝึกหัดที่ 3.3 : เมธอด PrintCharLine

เมธอด PrintCharLine ด้านล่างนี้ดัดแปลงมาจากเมธอด PrintLine ในแบบฝึกหัดที่ 3.2

เมธอดใหม่นี้กำหนดให้มีพารามิเตอร์สองตัว คือ c เป็นชนิด char และ len เป็นชนิด int โดยที่พารามิเตอร์ c ใช้สำหรับระบุอักขระที่จะทำการพิมพ์ออกทางจอภาพ (ไม่ได้พิมพ์แค่ดาวอย่างเดียวอีกต่อไป) และ len ใช้ระบุจำนวนอักขระที่ต้องการพิมพ์ในหนึ่งบรรทัด

```
using  System;
class  ParamNoRet
{
        static  void  PrintCharLine(char  c,  int  len)
        {
              int i = 0;
              while(i  <  len)
              {
                 Console.Write(c);
                 i++;
              }
              Console.WriteLine();
        }
        static  void  Main()
        {
               PrintCharLine('o',  10);
               PrintCharLine('x',  20);
        }
}
```

ทดลองพิมพ์และรันโปรแกรมในตัวอย่าง โปรแกรมแสดงผลลัพธ์อย่างไร

จงดัดแปลงเมธอก Main เพื่อให้โปรแกรมแสดงอักขระ 'x' จำนวน 10 ตัวในบรรทัดแรก อักขระ '*' จำนวน 20 ตัวในบรรทัดถัดมา และอักขระ 'v' จำนวน 30 ตัว ในบรรทัดสุดท้าย ดังนี้

```
xxxxxxxxxx
********************
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

เมธอด Main ที่เขียนได้คือ

`static void Main()`
`{`
 	
`}`

แบบฝึกหัดที่ 3.4 : สามเหลี่ยมของกากบาท

งานของคุณคือเขียนโปรแกรมที่รับขนาดแล้วพิมพ์สามเหลี่ยมจากตัว x ตามตัวอย่างต่อไปนี้

ตัวอย่างที่ 1	ตัวอย่างที่ 2	ตัวอย่างที่ 3	ตัวอย่างที่ 4
Enter height : **5** ----x ---xx --xxx -xxxx xxxxx	Enter height : **3** --x -xx xxx	Enter height : **1** x	Enter height : **0**

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)

```
using System;
class MainClass
{
    public static void PrintChar(char c, int len)
    {
        int i = 0;
        while(i < len)
        {
            Console.Write(c);
            i++;
        }
    }

    public static void Main(string[] args)
    {
        Console.Write("Enter height : ");
        int h = int.Parse(Console.ReadLine());
        int i = 0;
        while(i < h)
        {
            PrintChar('-',___________________);
            PrintChar('x',___________________);
            Console.WriteLine();
            i++;
        }
    }
}
```

แบบฝึกหัดที่ 3.5 : สามเหลี่ยมของกากบาท กลับหัว

งานของคุณคือเติมโปรแกรมด้านล่างที่พิมพ์สามเหลี่ยมจากตัว x ให้สมบูรณ์ ตามตัวอย่างด้านล่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	ตัวอย่างที่ 1 Enter height : 5 xxxxx -xxxx --xxx ---xx ----x _____ **ตัวอย่างที่ 2** Enter height : 3 xxx -xx --x

แบบฝึกหัดที่ 3.6 : เมธอด PlotChar

เราจะทดสอบสร้างเมธอดอีกอันหนึ่งชื่อ PlotChar เพื่อนำไปใช้ในโจทย์ฝึกโปรแกรมในตอนท้าย เมธอดนี้รับพารามิเตอร์สองตัว คือ c เป็นชนิด char และ dist เป็นชนิด int ซึ่งคล้ายคลึงกับเมธอด PrintCharLine ที่ผ่านมา แต่เมธอด PlotChar จะทำการพิมพ์ช่องว่างเป็นจำนวน dist - 1 ตัวอักษร และทำการพิมพ์อักขระในพารามิเตอร์ c ปิดท้ายเพียงตัวเดียวพร้อมทั้งขึ้นบรรทัดใหม่

ให้เติมคำสั่งลงในช่องว่าง เพื่อให้โปรแกรมแสดงผลตามผลลัพธ์ที่แสดง ๆ ดังต่อไปนี้

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน

using System;
class Stars {
static void PlotChar(char c, int dist)
{
_____
_____
_____
_____
}
 	
static void Main() {
PlotChar('x',1);
PlotChar('-',2);
PlotChar('+',3);
PlotChar('o',6);
}
}
	x
	-
	+
	o

แบบฝึกหัดที่ 3.7 : ลำดับฮาร์โมนิค

จงเขียนเมธอด f ที่รับพารามิเตอร์ n จากนั้นคำนวณค่า H_n เมื่อ H_n มีนิยามดังนี้

$$H_n = \sum_{i=1}^{n} \frac{1}{i}$$

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน

`using System;`
 	
`class x`
`{`
`  _____`
`  _____`
`  _____`
`  _____`
`  _____`
 	
`  static void Main()`
`  {`
`  Console.WriteLine(" n | f(n)");`
`  Console.WriteLine("---+------");`
`  int n = 1;`
`  while(n <= 15)`
`  {`
`  Console.WriteLine("{0,2}  |  {1:f3}",  n,  f(n));`
`  n++;`
`  }`
`  }`
`}`
	`n | f(n)`
	`---+------`
	`1 | 1.000`
	`2 | 1.500`
	`3 | 1.833`
	`4 | 2.083`
	`5 | 2.283`
	`6 | 2.450`
	`7 | 2.593`
	`8 | 2.718`
	`9 | 2.829`
	`10 | 2.929`
	`11 | 3.020`
	`12 | 3.103`
	`13 | 3.180`
	`14 | 3.252`
	`15 | 3.318`

แบบฝึกหัดที่ 3.8 : เมธอดคำนวณแฟคทอเรียล

เขียนเมธอด Factorial ที่ประกาศด้านล่างให้สมบูรณ์ เมธอดดังกล่าวรับจำนวนเต็ม n ที่มีค่ามากกว่าเท่ากับ 0 จากนั้นคืนค่าแฟคทอเรียลของ n

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน

`using System;`
 	
`class x`
`{`
`  static int Factorial(int n)`
`  {`
`  _____`
`  _____`
`  _____`
`  _____`
`  _____`
`  }`
 	
`  static void Main()`
`  {`
`  Console.Write("Enter N: ");`
`  int n = int.Parse(Console.ReadLine());`
`  Console.WriteLine(Factorial(n));`
`  }`
`}`
	ตัวอย่างที่ 1
	`Enter N: 0`
	`1`
	_____
	ตัวอย่างที่ 2
	`Enter N: 3`
	`6`
	_____
	ตัวอย่างที่ 3
	`Enter N: 7`
	`5040`

แบบฝึกหัดที่ 3.9 : ประมาณค่า

ค่าคงที่ e ในทางคณิตศาสตร์ มีค่าเท่ากับ $\sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots$

ในข้อนี้เราจะประมาณค่า โดยโปรแกรมจะรับค่า k แล้วคำนวณ $\sum_{n=0}^{k} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{k!}$

(นั่นคือ k + 1 พจน์แรกของนิพจน์ข้างต้น)

ให้เขียนเมธอด Estimate ที่รับพารามิเตอร์ k จากนั้นคำนวณค่าประมาณของ ในการเขียนเมธอด Estimate สามารถเรียกใช้เมธอด Factorial ในแบบฝึกหัดที่ 3.8 ได้โดยไม่ต้องเขียนใหม่ (แต่ในโปรแกรมของนิสิตจะต้องมีเมธอด Factorial ที่สมบูรณ์)

โปรแกรมที่เขียนได้คือ

```
using System;

class x
{
  static int Factorial(int n)
  {
    /* โปรแกรมของเมธอด Factorial ที่ได้เขียนไว้แล้วในแบบฝึกหัดที่ 3.8  */
  }
  static double Estimate(int k)
  {
     ____________________________________
     ____________________________________
     ____________________________________
  }
  static void Main()
  {
    Console.Write("Enter k: ");
    int k = int.Parse(Console.ReadLine());
    Console.WriteLine("{0:f5}",Estimate(k));
  }
}
```

ตัวอย่างการทำงาน

ตัวอย่างที่ 1

```
Enter N: 1
2.00000
```

ตัวอย่างที่ 2

```
Enter N: 3
2.66667
```

ตัวอย่างที่ 3

```
Enter N: 7
2.71825
```

แบบฝึกหัดที่ 3.10 : กราฟ sine

เขียนโปรแกรมเพื่อวาดกราฟอย่างง่ายของฟังชัน $f(x) = 20 + 10 \sin \frac{x}{\pi}$

โดยที่ x มีค่าตั้งแต่ 0 ถึง 20 และฟังกชัน sin(x) ให้ค่าซายน์ของมุม x เรเดียนโปรแกรมที่สมบูรณ์ควรแสดงผลลัพธ์ (หมายเหตุ ในการแสดงผล ให้ค่า x มีค่าเท่ากับจำนวนเต็มที่ใกล้เคียงกับค่า x ที่สุด (ใช้ฟังก์ชัน Math.Round()))

โปรแกรมที่เขียนได้คือ

ตัวอย่างการทำงาน

```
 x|
--+---------------------------------------> y
 0|                    *
 1|                       *
 2|                         *
 3|                            *
 4|                             *
 5|                             *
 6|                             *
 7|                           *
 8|                         *
 9|                      *
10|                   *
11|               *
12|            *
13|          *
14|        *
15|        *
16|        *
17|          *
18|             *
19|                *
20|                   *
```

แบบฝึกหัดที่ 3.11 : วัตถุสองแรง

มีวัตถุมวล m กิโลกรัม รูปลูกบาศก์ วางอยู่บนพื้นราบที่ไม่มีแรงเสียดทาน มีแรงกระทำกับวัตถุสองแรง

- แรงแรก f1 กระทำกับวัตถุดังกล่าว แรงกระทำไปทางทิศทางขวา โดยมีทิศทำมุม t1 องศากับแนวระนาบ

- แรงที่สอง f2 กระทำกับวัตถุดังกล่าว แรงกระทำไปทางทิศทางซ้ายโดยมีทิศทำมุม t2 องศากับแนวระนาบ

ให้นักเรียนเขียนเมธอด read_m() สำหรับอ่านปริมาณมวลของวัตถุ, read_f() สำหรับอ่านปริมาณแรง, read_t() สำหรับอ่านขนาดของมุมโดยคืนค่าของมุมเป็นเรเดียน, calculate_a() สำหรับคำนวณความเร่งของวัตถุ และ show_output() เพื่อแสดงข้อมูลความเร่งที่คำนวณได้ โดยคิดทิศทางไปทางขวาเป็นบวก และโปรแกรมมีเมธอด Main() เป็นดังนี้

เมธอด Main	ตัวอย่างการทำงาน

`using System;`
`class MainClass`
`{`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`_____`
`static void Main ()`
`{`
`double m, f1, t1, f2, t2, a;`
`m = read_m();`
`f1 = read_f(1);`
`t1 = read_t(1);`
`f2 = read_f(2);`
`t2 = read_t(2);`
`a = calculate_a(m,f1,t1,f2,t2);`
`show_output(a);`
`}`
`}`
	`Enter m: 1`
	`Enter f1: 10`
	`Enter t1: 45`
	`Enter f2: 5`
	`Enter t2: 10`
	`a = 2.1470`