

Lab 8 - รวมคำสั่งภาษา C# พื้นฐาน

1. โครงสร้างของภาษา C#

เราได้เรียนรู้แล้วว่าโครงสร้างพื้นฐานของภาษา C# นั้นต้องประกอบไปด้วย เนมสเปซ (namespace, จะมีหรือไม่มีก็ได้ขึ้นอยู่กับแต่ละโปรแกรม) คลาส (class, ต้องมีอย่างน้อยหนึ่งคลาสเสมอ) และเมธอด (method, มีเมธอด Main เป็นอย่างน้อย แต่อาจจะมีเมธอดอื่น ๆ เพิ่มเติมอีกทั้งนี้ขึ้นอยู่กับแต่ละโปรแกรมเช่นกัน) ดังภาพที่ 1.1

```
namespace something
{
    class example
    {
        static void Main()
        {
            // เริ่มต้นเขียนโปรแกรมที่จุดนี้
        }
    }
}
```

ภาพที่ 1.1 : โครงสร้างของภาษา C#

2. คำสั่งแสดงผล (Output statements)

คำสั่งที่ใช้ในการแสดงผลของภาษา C# ที่เราได้ศึกษาไปบ้างแล้วใน Lab 7 คือ คำสั่ง Console.Write() และ คำสั่ง Console.WriteLine() ซึ่งความแตกต่างคือ Console.Write() จะไม่ทำการขึ้นบรรทัดใหม่ แต่ Console.Write() จะขึ้นบรรทัดใหม่ให้หลังแสดงผล ทั้งนี้คำสั่ง Console.Write() และ Console.WriteLine() สามารถใช้งานได้หลากหลายรูปแบบดังนี้

แบบที่ 1 Console.Write(<ข้อความ>); หรือ Console.WriteLine(<ข้อความ>); ผลลัพธ์ที่ได้คือโปรแกรมจะนำข้อความไปแสดงผลบนหน้าจอคอมพิวเตอร์

ตัวอย่างการใช้งาน

```
Console.Write("Hello!!");
```

แบบที่ 2 Console.Write(<ตัวแปร>); หรือ Console.WriteLine(<ตัวแปร>); ผลลัพธ์ที่ได้คือโปรแกรมจะนำค่าที่เก็บอยู่ในตัวแปรนั้น ๆ ไปแสดงผลบนหน้าจอคอมพิวเตอร์

ตัวอย่างการใช้งาน

```
int x = 5;
Console.Write(x);
```

แบบที่ 3 Console.Write(<ข้อความที่มีสัญลักษณ์ {0}, {1}, {2}, ... แทรกอยู่>, นิพจน์0, นิพจน์1, นิพจน์2, ...); หรือ Console.WriteLine(<ข้อความที่มีสัญลักษณ์ {0}, {1}, {2}, ... แทรกอยู่>, นิพจน์0, นิพจน์1, นิพจน์2, ...); ผลลัพธ์ที่ได้คือโปรแกรมจะแสดงข้อความนั้น ๆ บนหน้าจอคอมพิวเตอร์แต่ส่วนที่เป็น {0}, {1}, {2}, ... จะถูกแทนที่ด้วยค่าที่คำนวณได้จากนิพจน์0, นิพจน์1, นิพจน์2, ... ตามลำดับ

ตัวอย่างการใช้งาน

```
int score = 9; string name = "Mc";
Console.WriteLine("{0}'s score is {1}", name, score);
```

แบบฝึกหัดที่ 1.1 : WriteLine, Gandhi

เขียนโปรแกรมด้านล่างให้สมบูรณ์ ให้โปรแกรมพิมพ์ผลลัพธ์เป็นข้อความดังต่อไปนี้

```
Live as if you were to die tomorrow.
Learn as if you were to live forever.
-Mahatma Gandhi
```

โปรแกรมที่กำหนดให้ (เดิมโปรแกรมในช่องว่างให้สมบูรณ์)

```
using System;
class MainClass
{
    public static void Main(string[] args)
    {
        _____
        _____
        _____
    }
}
```

แบบฝึกหัดที่ 1.2 : WriteLine ที่มีตัวแปร

เขียนโปรแกรมด้านล่างให้สมบูรณ์ ให้โปรแกรมพิมพ์ผลลัพธ์เป็นข้อความดังต่อไปนี้

```
Hello Arthur. You are 18 years old.
```

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)

```
using System;

class SayHi {
    static void Main() {
        string yourName = _____;
        int yourAge = _____;
        Console.WriteLine("Hello {1}. You are {0} years old.", _____, _____);
    }
}
```

2. ตัวแปรในภาษา C# (Variable in C#)

ตัวแปรในภาษา C# ต่างจากภาษา Python ตรงที่ชนิดของตัวแปรมีความสำคัญมาก และเราจำเป็นต้องประกาศก่อนใช้งานทุกครั้ง ชนิดของตัวแปรในภาษา C# ที่ใช้บ่อย ๆ มีดังนี้

ชนิดของตัวแปร	ข้อมูลที่เก็บได้
int	จำนวนเต็ม
double	จำนวนจริง
bool	ค่าทางตรรกศาสตร์ (จริง / เท็จ)
char	ตัวอักษรตัวเดียว
string	ข้อความ (ตัวอักษรหลายตัว)

การประกาศตัวแปรมีรูปแบบดังนี้

```
<ชนิดของตัวแปร> <ชื่อตัวแปร>;
```

ตัวอย่างการประกาศตัวแปร name ที่มีชนิดข้อมูลเป็น string ทำได้ดังนี้

```
string name;
```

ทั้งนี้เราสามารถใส่ค่าเริ่มต้นให้กับตัวแปรได้ (แต่เป็นค่าเริ่มต้นสามารถเปลี่ยนแปลงค่าภายหลังได้ถ้าต้องการ) โดยมีรูปแบบดังนี้

```
<ชนิดของตัวแปร> <ชื่อตัวแปร> = <ค่าเริ่มต้น>;
```

ตัวอย่างการประกาศตัวแปร rate ที่มีชนิดข้อมูลเป็น double และมีค่าเริ่มต้นเป็น 0.75 ทำได้ดังนี้

```
double rate = 0.75;
```

แบบฝึกหัดที่ 2.1 : จงประกาศตัวแปรที่สอดคล้องกับข้อกำหนดต่อไปนี้

ชนิดของตัวแปร	ชื่อตัวแปร	ค่าเริ่มต้น	การประกาศตัวแปรที่เขียนได้คือ
ข้อความ	password	"Secretcode"	
จำนวนเต็ม	point	ไม่มีค่าเริ่มต้น	
ตัวอักษรตัวเดียว	gender	'M'	
ค่าทางตรรกศาสตร์	isright	true	

3. คำสั่งรับข้อมูลเข้า (Input statement)

คำสั่งที่ใช้รับข้อมูลเข้าสู่โปรแกรมผ่านแป้นพิมพ์ในภาษา C# คือคำสั่ง Console.ReadLine() โดยโปรแกรมจะรับข้อมูลเป็นชนิดข้อความ (ฉะนั้นหากต้องการคำนวณเป็นข้อมูลชนิดอื่น ๆ จำเป็นต้องใช้คำสั่งอื่น ๆ เพิ่มเติมเพื่อแปลงชนิดของข้อมูลก่อน) ดังตัวอย่าง

```
string input;
Console.Write("Please input something : ");
input = Console.ReadLine();
Console.WriteLine(input);
```

แบบฝึกหัดที่ 3.1 : สวัสดี

เขียนโปรแกรมที่อ่านชื่อผู้ใช้ จากนั้นแสดงคำทักทาย

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter your name: <u>Somchai</u> Hello, Somchai

4. การแปลงชนิดข้อมูลจากชนิดข้อความเป็นชนิดอื่น ๆ

จากหัวข้อที่ 3 เราพบว่าข้อมูลที่ได้รับผ่านคำสั่ง Console.ReadLine() นั้นจะได้ชนิดข้อมูลเป็นข้อความเสมอ ฉะนั้นหากเราต้องการนำไปคำนวณ บวก ลบ คูณ หรือหาร อื่น ๆ ต่อ อาจได้ข้อมูลที่ผิดพลาดได้ เราจึงต้องใช้คำสั่งในการแปลงชนิดข้อมูลจากชนิดข้อความเป็นชนิดอื่น ๆ ที่เหมาะสมดังนี้

คำสั่ง	การทำงาน	ตัวอย่าง
int.Parse()	แปลงชนิดข้อมูลจากข้อความเป็นจำนวนเต็ม	int.Parse("10")
double.Parse()	แปลงชนิดข้อมูลจากข้อความเป็นจำนวนจริง	double.Parse("5.2")
char.Parse()	แปลงชนิดข้อมูลจากข้อความเป็นตัวอักษรตัวเดียว	char.Parse("M")

ดังนั้นการรับข้อมูลที่ต้องการได้ข้อมูลเป็นเลขจำนวนเต็มจึงสามารถทำได้ดังนี้

```
string input; int n;
Console.Write("Please input n : ");
input = Console.ReadLine();
n = int.Parse(input);
```

หรือสามารถย่อได้เป็น

```
int n;
Console.Write("Please input n : ");
n = int.Parse(Console.ReadLine());
```

5. การดำเนินการทางคณิตศาสตร์

การดำเนินการทางคณิตศาสตร์ในภาษา C# นั้นสำหรับเครื่องหมาย บวก (+) ลบ (-) คูณ (*) และหารเอาเศษ (%) นั้นเหมือนกับภาษาไพทอนทุกประการแต่นักเรียนควรต้องระวังในการใช้ก็คือเครื่องหมายหาร (/) เนื่องจากชนิดของข้อมูลจะส่งผลถึงผลที่ได้จากการหารด้วยดังตารางต่อไปนี้

รูปแบบของการหาร	ตัวอย่าง	ผลลัพธ์ที่ได้ในภาษา C#	ผลลัพธ์ที่ได้ในภาษาไพทอน
จำนวนเต็ม / จำนวนเต็ม	45 / 8	5	5.625
จำนวนจริง / จำนวนเต็ม	45.0 / 8	5.625	5.625
จำนวนเต็ม / จำนวนจริง	45 / 8.0	5.625	5.625
จำนวนจริง / จำนวนจริง	45.0 / 8.0	5.625	5.625

แบบฝึกหัดที่ 5.1 : การหารจำนวนเต็ม

เขียนโปรแกรมที่รับ input เป็นเลขจำนวนเต็ม 2 จำนวน ประกอบด้วยตัวตั้ง (Dividend) กับตัวหาร (Divisor) โดยโปรแกรมจะแสดง output เป็นผลหารและเศษที่เหลือจากผลหาร

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<pre>Input Dividend : <u>35</u> Input Divisor : <u>4</u> The quotient of 35/4 is 8 and the remainder is 3</pre>

แบบฝึกหัดที่ 5.2 : แปลงอุณหภูมิ

จงเขียนโปรแกรมที่รับอุณหภูมิในหน่วยองศาเซลเซียสเป็นจำนวนจริง จากนั้นแปลงอุณหภูมิดังกล่าวเป็นฟาเรนไฮต์ โดยโปรแกรมจะคำนวณตามสูตร

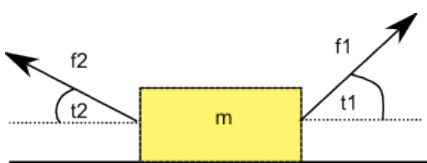
$$Fahrenheit = \frac{9}{5} \times (Celsius) + 32$$

*หมายเหตุ ในการแสดงค่าอุณหภูมิให้แสดงเป็นทศนิยม 1 ตำแหน่ง (ใช้รูปแบบ :f1 ในการพิมพ์ด้วยคำสั่ง Console.WriteLine นั่นคือ แทนที่จะใช้ {0} ให้ใช้ {0:f1} แทน ในกรณีที่ป้อนพารามิเตอร์แรกที่พิมพ์)

ตัวอย่างการทำงาน
<p>ตัวอย่างที่ 1</p> <pre>Input a temperature in degree Celsius : <u>28</u> 28.0 degrees Celsius is equivalent to 82.4 degrees Fahrenheit</pre> <hr/> <p>ตัวอย่างที่ 2</p> <pre>Input a temperature in degree Celsius : <u>20.1</u> 20.1 degrees Celsius is equivalent to 68.2 degrees Fahrenheit</pre>

โปรแกรมที่เขียนได้คือ

แบบฝึกหัดที่ 5.3 : วัตถุสองแรง



มีวัตถุมวล m กิโลกรัม รูปลูกบาศก์ วางอยู่บนพื้นราบที่ไม่มีแรงเสียดทาน มีแรงกระทำกับวัตถุสองแรง

- แรงแรก f_1 กระทำกับวัตถุดังกล่าว แรงกระทำไปทางทิศทางขวา โดยมีทิศทำมุม t_1 องศา กับแนวระนาบ
- แรงที่สอง f_2 กระทำกับวัตถุดังกล่าว แรงกระทำไปทางทิศทางซ้าย โดยมีทิศทำมุม t_2 องศา กับแนวระนาบ

โดยมีทิศทำมุม t_2 องศา กับแนวระนาบ

จงเขียนโปรแกรมเพื่อคำนวณหาความเร่งในแนวระนาบของวัตถุดังกล่าว ให้คิดทิศทางไปทางขวาเป็นบวก

* หมายเหตุ ในการแสดงผลให้พิมพ์เป็นเลขทศนิยม 4 ตำแหน่ง โดยให้ใช้ {0:F4} แทน {0} เมื่อสั่งพิมพ์ด้วย Console.WriteLine

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<pre>Enter m: <u>1</u> Enter f1: <u>10</u> Enter t1: <u>45</u> Enter f2: <u>5</u> Enter t2: <u>10</u> a = 2.1470</pre>

6. นิพจน์ทางตรรกศาสตร์ (Boolean Expression)

เมื่อเราต้องการเปรียบเทียบเพื่อหาค่าจริงหรือเท็จในภาษา C# สามารถใช้เครื่องหมาย >, <, >=, <=, == และ != ได้เหมือนที่ใช้ในภาษาไพทอนทุกประการ แต่หากต้องการเชื่อมประพจน์เหล่านั้นเข้าด้วยกันด้วยคำเชื่อม and, or หรือ not ต้องปรับรูปแบบใหม่ดังตาราง

ตัวเชื่อมทางตรรกศาสตร์ในภาษาไพทอน	ตัวเชื่อมทางตรรกศาสตร์ในภาษา C#
and	&&
or	
not	!

7. คำสั่งเงื่อนไข (If statements)

คำสั่ง if และ if - else ในภาษา C# นั้นก็มีการทำงานเหมือนกันคำสั่ง if และ if - else ในภาษาไพทอนทุกประการ ต่างเพียงรูปแบบเพียงเล็กน้อยเท่านั้น และภาษา C# ไม่มีคำสั่ง elif เหมือนในภาษาไพทอน เราจึงจำเป็นต้องเขียน else if แบบเต็ม ๆ แทน ขอให้นักเรียนพิจารณาจากตารางที่ 7.1

เลขบรรทัด	คำสั่งในภาษาไพทอน	คำสั่งในภาษา C#
1	if condition1 :	if (condition1)
2	statement1	{
3	statement2	statement1
4	elif condition2 :	statement2
5	statement3	}
6	else :	else if (condition2)
7	statement4	statement3
8	statement5	else
9		{
10		statement4
11		statement5
12		}

ตารางที่ 7.1 : ตารางเทียบรูปแบบการใช้คำสั่ง if - else ใน ภาษาไพทอน และ ภาษา C#

หากนักเรียนสังเกตคำสั่ง statement3 ใด ๆ จะพบว่าคำสั่ง statement3 เป็นคำสั่งเพียงคำสั่งเดียวที่อยู่ในความควบคุมของเงื่อนไขที่สอง (condition2) ในภาษา C# หากในเงื่อนไขใดที่คำสั่ง (statement) เพียงคำสั่งเดียว ผู้เขียนโปรแกรมอาจไม่ต้องใส่เครื่องหมาย { } ครอบเพื่อระบุขอบเขตของความควบคุมก็ได้ (ทั้งนี้จะไม่ใส่ก็ได้เช่นเดียวกัน) ทำให้ในภาษา C# เราไม่จำเป็นต้องใส่เครื่องหมาย { และ } แทรกระหว่างบรรทัดที่ 6 - 7 และ 7 - 8 ตามลำดับ (ทั้งนี้หากนักเรียนคนใดกลัวงง จะใส่ { } ก็ได้ โปรแกรมก็จะสามารถทำงานได้ตามปกติ

แบบฝึกหัดที่ 7.1 : อายุเลือกตั้ง

จงเติมส่วนของโปรแกรมต่อไปนี้เพื่อให้เป็นโปรแกรมที่อ่านอายุของผู้ใช้ จากนั้นตรวจสอบว่าผู้ใช้สามารถเลือกตั้งได้หรือยัง ทั้งนี้กฎหมายระบุว่าบุคคลจะลงคะแนนได้ถ้าอายุมากกว่าหรือเท่ากับ 18 ปี

จงเติมส่วนของโปรแกรมต่อไปนี้ให้สมบูรณ์
<pre>using System; class Test { static void Main() { Console.WriteLine("Enter your age: "); int age = _____ if(_____) Console.WriteLine("You cannot vote."); else Console.WriteLine("You can vote. Exercise your right!"); } }</pre>

แบบฝึกหัดที่ 7.2 : ออกเกรดแบบง่าย

มีการสอบครั้งหนึ่ง คะแนนเต็มคือ 100 คะแนน จงเติมส่วนของโปรแกรมต่อไปนี้ให้เป็นโปรแกรมที่อ่านคะแนนแล้วระบุเกรดตามเงื่อนไขต่อไปนี้

- ถ้าคะแนนมีค่าอย่างน้อย 80, โปรแกรมจะต้องพิมพ์ Excellent job.
- ถ้าคะแนนน้อยกว่า 80 แต่มากกว่าหรือเท่ากับ 50, โปรแกรมพิมพ์ Okay.
- ถ้าไม่เช่นนั้น ให้โปรแกรมพิมพ์ว่า Please try harder.

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
<pre>using System; class Test { static void Main() { Console.WriteLine("Enter the score: "); int s = _____ if(_____) Console.WriteLine("Excellent job."); else if(_____) Console.WriteLine("Okay."); else _____ } }</pre>	<p>ตัวอย่างที่ 1 Enter the score: <u>85</u> Excellent job. _____</p> <p>ตัวอย่างที่ 2 Enter the score: <u>45</u> Please try harder. _____</p> <p>ตัวอย่างที่ 3 Enter the score: <u>60</u> Okay.</p>

แบบฝึกหัดที่ 7.3 : ตรวจสอบเขต

เขียนโปรแกรมที่อ่านจำนวนเต็มหนึ่งจำนวน จากนั้นตรวจสอบว่าจำนวนนั้นอยู่ในขอบเขต 100 ถึง 200 หรือไม่ (รวม 100 และ 200 ด้วย) ตัวอย่างเช่น 100, 150, 200 อยู่ในขอบเขต ส่วน 99, 1000 หรือ 201 จะไม่อยู่ในขอบเขต

* ค่าไป ต้องใช้ && ในการเชื่อมเงื่อนไข

จงเติมส่วนของโปรแกรมต่อไปนี้ให้สมบูรณ์
<pre>using System; class Test { static void Main() { int n = if(_____) Console.WriteLine("{0} is in the range.", n); else Console.WriteLine("Sorry. It is not in range."); } }</pre>

แบบฝึกหัดที่ 7.4 : ขนาดผลไม้

ลูกทุเรียนถูกแบ่งขนาดออกเป็น 3 กลุ่มตามน้ำหนัก ดังนี้

- น้อยกว่า 1 กิโลกรัม เรียกว่าเป็นขนาดเล็ก (small)
- มากกว่าหรือเท่ากับ 1 กิโลกรัม แต่น้อยกว่า 3 กิโลกรัม เรียกว่าเป็นขนาดกลาง (medium)
- มากกว่าหรือเท่ากับ 3 กิโลกรัม เรียกว่าเป็นขนาดใหญ่ (large)

ให้นักเรียนเขียนโปรแกรมเพื่อคัดขนาดของทุเรียนดังตัวอย่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Enter weight: <u>1.5</u> It is medium.</p> <p>_____</p> <p>ตัวอย่างที่ 2 Enter weight: <u>3</u> It is large.</p> <p>_____</p> <p>ตัวอย่างที่ 3 Enter weight: <u>0.5</u> It is small.</p>

แบบฝึกหัดที่ 7.5 : ฟังก์ชันแอสซูก

ให้นักเรียนเขียนโปรแกรมเพื่อคำนวณหาค่า $f(x)$ จากสมการดังต่อไปนี้

$$f(x) = \begin{cases} 2x + 10, & x \leq 15 \\ 3x^2, & 15 < x \leq 35 \\ 2x^3 - 5, & x > 35 \end{cases}$$

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Please input x: <u>10</u> $f(x) = 30.000$</p> <p>_____</p> <p>ตัวอย่างที่ 2 Please input x: <u>20</u> $f(x) = 1200.000$</p> <p>_____</p> <p>ตัวอย่างที่ 3 Please input x: <u>40</u> $f(x) = 127995.000$</p>

8. ฟังก์ชันทางคณิตศาสตร์ (Math Methods)

ภาษา C# มีฟังก์ชันพิเศษเกี่ยวกับคณิตศาสตร์เช่นเดียวกับภาษาไพทอน แต่ในภาษา C# เราจะเรียกว่าเมธอด ดังตัวอย่างเมธอดทางคณิตศาสตร์ดังต่อไปนี้

เมธอด	การทำงาน	ตัวอย่าง	ผลลัพธ์ที่ได้
Math.PI	ค่าของ π	Math.PI	3.1415927
Math.Min(x,y)	คืนค่าที่น้อยที่สุดระหว่าง x กับ y	Math.Min(10, 45)	10
Math.Max(x,y)	คืนค่าที่มากที่สุดระหว่าง x กับ y	Math.Max(10, 45)	45
Math.Pow(x,y)	คืนค่า x^y	Math.Pow(2, 3)	8
Math.Abs(x)	คืนค่าสัมบูรณ์ของ x	Math.Abs(-3.4)	3.4
Math.Sqrt(x)	คืนค่ารากที่สองของ x	Math.Sqrt(4.0)	2.0
Math.Floor(x)	คืนค่าจำนวนเต็มที่ยกที่สุดที่น้อยกว่าหรือเท่ากับ x	Math.Floor(3.9)	4.0
Math.Ceiling(x)	คืนค่าจำนวนเต็มที่ยกที่สุดที่มากกว่าหรือเท่ากับ x	Math.Ceiling(6.1)	6.0
Math.Round(x)	คืนค่าจำนวนเต็มที่ใกล้เคียงกับค่า x มากที่สุด	Math.Round(1.3)	1.0

แบบฝึกหัดที่ 8.1 : ไปกินไอติม

คุณทำงานหาเงินมาได้ทั้งสิ้น 3 วัน ในแต่ละวันคุณเก็บเศษสตางค์ที่ได้มารวมกัน ถ้าเศษสตางค์นี้รวมกันได้มากกว่า 1 บาท คุณจะนำเงินที่ได้ไปซื้อไอศกรีม จงเขียนโปรแกรมเพื่อตรวจสอบเงื่อนไขดังกล่าว

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
<pre>using System; class Program { public static void Main(string[] args) { double s1 = double.Parse(Console.ReadLine()); double s2 = double.Parse(Console.ReadLine()); double s3 = double.Parse(Console.ReadLine()); double t1 = _____ double t2 = _____ double t3 = _____ if(_____) Console.WriteLine("I can have an icecream."); else Console.WriteLine("I will have to wait."); } }</pre>	<p>ตัวอย่างที่ 1</p> <p><u>10.50</u> <u>3.40</u> <u>0.30</u> I can have an icecream.</p> <hr/> <p>ตัวอย่างที่ 2</p> <p><u>10.10</u> <u>3.40</u> <u>100.05</u> I will have to wait.</p> <hr/> <p>ตัวอย่างที่ 3</p> <p><u>1212.12</u> <u>31.80</u> <u>11.09</u> I can have an icecream.</p>

9. คำสั่งวนซ้ำแบบ while (while statement)

ในภาษา C# มีคำสั่งแบบ while เช่นเดียวกับภาษาไพทอน โดยแตกต่างกันที่รูปแบบเพียงเล็กน้อยเช่นเดียวกัน ขอให้นักเรียนพิจารณาจากตารางต่อไปนี้

รูปแบบคำสั่ง while ในภาษาไพทอน	รูปแบบคำสั่ง while ในภาษา C#
while condition : statement1 statement2 statement3	while (condition) { statement1 statement2 statement3 }

แบบฝึกหัดที่ 9.1 : ไล่เลข

จงเขียนโปรแกรมรับจำนวนเต็ม n จากนั้นให้โปรแกรมพิมพ์จำนวนเต็มตั้งแต่ 1 ถึง n บรรทัดละหนึ่งตัว

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
<pre>using System; class SimpleWhile { static void Main(string [] argv) { _____ int n = _____ int i = 1; while(_____) { Console.WriteLine(_____); _____ } } }</pre>	<p>Enter N: <u>5</u></p> <p>1 2 3 4 5</p>

แบบฝึกหัดที่ 9.2 : ผลรวม 1

เขียนโปรแกรมรับจำนวนเต็มเพื่อหาผลรวม โดยโปรแกรมจะรับจำนวนเต็มแล้วนำไปรวมเป็นผลรวมไปเรื่อย ๆ จนกระทั่งผู้ใช้พิมพ์ค่าที่เป็นลบ สำหรับข้อนี้ให้รวมค่าสุดท้ายดังกล่าวในผลรวมด้วย

ทั้งนี้ในแต่ละรอบที่คำนวณหาผลรวม ให้โปรแกรมพิมพ์ผลรวมที่ได้เมื่อรับข้อมูลแต่ละตัวด้วยดังตัวอย่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
<pre>using System; class Program { public static void Main(string[] args) { int total = _____; int data = _____; while(_____) { Console.Write(_____); data = _____ _____ Console.WriteLine("Total = {0}",total); } Console.ReadLine(); } }</pre>	<p>Enter data: <u>10</u> Total = 10 Enter data: <u>23</u> Total = 33 Enter data: <u>-15</u> Total = 18</p>

แบบฝึกหัดที่ 9.3 : ผลรวม 2

เขียนโปรแกรมรับจำนวนเต็มเพื่อหาผลรวมเฉพาะข้อมูลที่เป็นเลขคู่ โดยโปรแกรมจะรับจำนวนเต็มแล้วนำไปรวมเป็นผลรวมไปเรื่อย ๆ จนกระทั่งผู้ใช้พิมพ์ค่าที่เป็นลบ ให้รวมค่าสุดท้ายดังกล่าวในผลรวมด้วยถ้าค่าดังกล่าวมีค่าเป็นเลขคู่ ทั้งนี้ในแต่ละรอบที่คำนวณหาผลรวม ให้โปรแกรมพิมพ์ผลรวมที่ได้เมื่อรับข้อมูลแต่ละตัวด้วยดังตัวอย่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1</p> <p>Enter data: <u>10</u> Total = 10 Enter data: <u>23</u> Total = 10 Enter data: <u>6</u> Total = 16 Enter data: <u>-15</u> Total = 16</p> <hr/> <p>ตัวอย่างที่ 2</p> <p>Enter data: <u>27</u> Total = 0 Enter data: <u>10</u> Total = 10 Enter data: <u>2</u> Total = 12 Enter data: <u>-8</u> Total = 4</p>

แบบฝึกหัดที่ 9.4 : รวมคะแนนสอบ

จงเขียนโปรแกรมที่รับคะแนนสอบแต่ละครั้งของนักเรียนคนหนึ่งเรื่อยๆ พร้อมทั้งแสดง คะแนนรวมทุกครั้ง โดยแสดงผลทศนิยม 1 ตำแหน่ง และจะหยุดรับคะแนนสอบเมื่อผู้ใช้พิมพ์คะแนนที่มีค่าเป็น -1 แล้วพิมพ์คำว่า PROGRAM TERMINATED!!! ดังตัวอย่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>Enter a score, or -1 to quit : <u>76</u> Now summation is : 76.0 Enter a score, or -1 to quit : <u>56.7</u> Now summation is : 132.7 Enter a score, or -1 to quit : <u>87.4</u> Now summation is : 220.1 Enter a score, or -1 to quit : <u>53.5</u> Now summation is : 273.6 Enter a score, or -1 to quit : <u>90.8</u> Now summation is : 364.4 Enter a score, or -1 to quit : <u>99</u> Now summation is : 463.4 Enter a score, or -1 to quit : <u>-1</u> PROGRAM TERMINATED!!!</p>

แบบฝึกหัดที่ 9.5 : พิมพ์เลขที่ละหลัก

เขียนโปรแกรมรับจำนวนเต็ม จากนั้นพิมพ์จำนวนเต็มดังกล่าวออกมาทีละหลัก หลักละ 1 บรรทัด โดยพิมพ์จากหลักหน่วยก่อน

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Enter number: <u>12437</u> 7 3 4 2 1 _____</p> <p>ตัวอย่างที่ 2 Enter number: <u>10000</u> 0 0 0 0 1 _____</p> <p>ตัวอย่างที่ 3 Enter number: <u>0</u> 0</p>

แบบฝึกหัดที่ 9.6 : แปลงเป็นฐาน 10

ให้เขียนรับตัวเลขฐาน 2 จากผู้ใช้ จากนั้นแปลงให้เป็นเลขฐานสิบ รับประกันว่าเลขฐานสองที่ป้อนจะมีจำนวนหลักไม่เกิน 9 หลัก

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Enter binary number: <u>10110</u> 22 _____</p> <p>ตัวอย่างที่ 2 Enter binary number: <u>110</u> 6</p>

Python – C# Comparison Sheet #1

Topics	Python	C#
Comments	<pre># A line beginning with '#' is a comment. """ In Python, you can use multiple-line strings as a multi-line comments. We have not learned this in class. """</pre>	<pre>// A line beginning with '//' is a comment /* If you want multiline comments, Put them in a pair of a slash-star and a star-slash. */</pre>
Types	int, float, str	int, double, string
Declaration	(no variable declaration)	<i>type variablename;</i>
Type conversion	<pre>int("100") → 100 int(5.4) → 5 float(10) → 10.0 str(10) → '10' str(10.5) → '10.5'</pre>	<pre>int.Parse("100") → 100 (int)(5.4) → 5 (double)(10) → 10.0 10.ToString() → '10' (10.5).ToString() → '10.5'</pre>
Input	<pre>a = input() x = int(input()) y = float(input())</pre>	<pre>string a = Console.ReadLine(); int x = int.Parse(Console.ReadLine()); double y = double.Parse(Console.ReadLine());</pre>
Input /w prompt	<pre>b = input("Enter B: ")</pre>	<pre>Console.WriteLine("Enter B: "); b = Console.ReadLine();</pre>
Output	<pre>print("hello, world") print('Cost = ',10,'baht')</pre>	<pre>Console.WriteLine("Hello, world"); Console.WriteLine("Cost = {0} baht",10);</pre>
Output/no newline	<pre>print("Hello, ",end='') print("world")</pre>	<pre>Console.WriteLine("Hello, "); Console.WriteLine("world");</pre>
Blocks (with examples of conditional "if" statements)	<pre># Python uses indentation if a < 10: print("hello") while b > 100: print(b) b -= 1 print("good bye")</pre>	<pre>// C# uses {} if(a < 10) { Console.WriteLine("hello"); while(b > 100) { Console.WriteLine(b); b--; } } Console.WriteLine("good bye");</pre>
If-else	<pre>if a > 10: print("Good") if a < 5: print("Bad") else: print("Normal")</pre>	<pre>if(a > 10) { Console.WriteLine("Good"); if(a < 5) Console.WriteLine("Bad"); } else Console.WriteLine("Normal");</pre>
While	<pre>i = 10 while i<=20: print(i) i += 1</pre>	<pre>int i = 10; while(i<=20) { Console.WriteLine(i); i++; }</pre>
Do-while	(no do-while loop)	<pre>int i = 10; do { Console.WriteLine(i); i--; } while(i > 0);</pre>
Random example	<pre>import math r = float(input("Enter radius: ")) print("The area is", math.pi * r * r)</pre>	<pre>Console.WriteLine("Enter radius: "); double r = double.Parse(Console.ReadLine()); Console.WriteLine("The area is {0}", Math.PI * r * r);</pre>

Note: Codes in shade are outside the scope of the course.

(Version 1.0)