Chapter 3

Flow Control & Exception

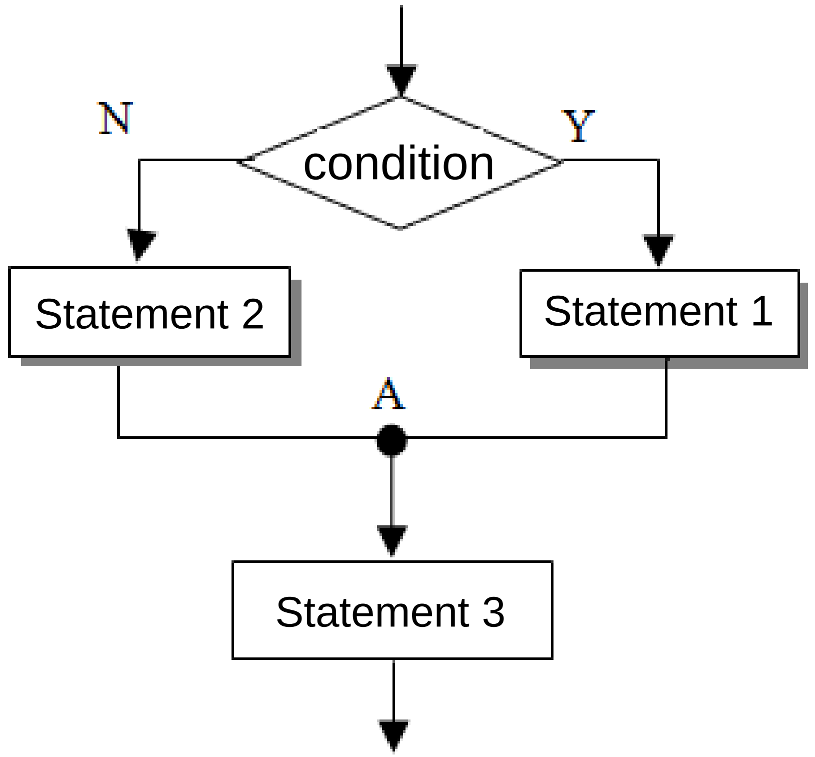
3.1 Selection Statements

3 selection statements in C#

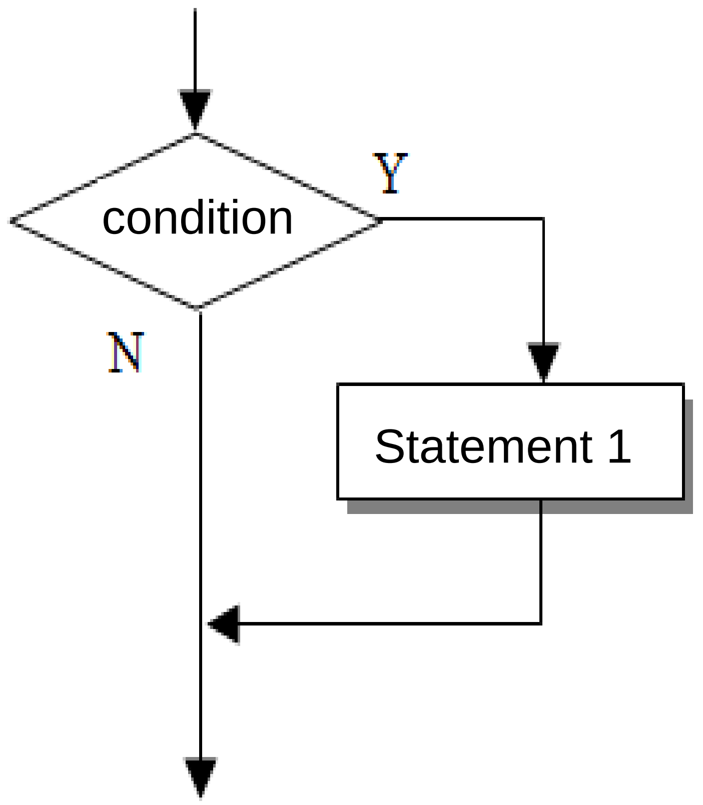
1. **if… else**
2. **if … else if … else**
3. **switch**

3.1.1 if…else

```
if ( [condition] )
{
    [statement 1]
}
else
{
    [statement 2]
}
[statement 3]
```



```
if ( [condition] )
{
    [statement 1]
}
```



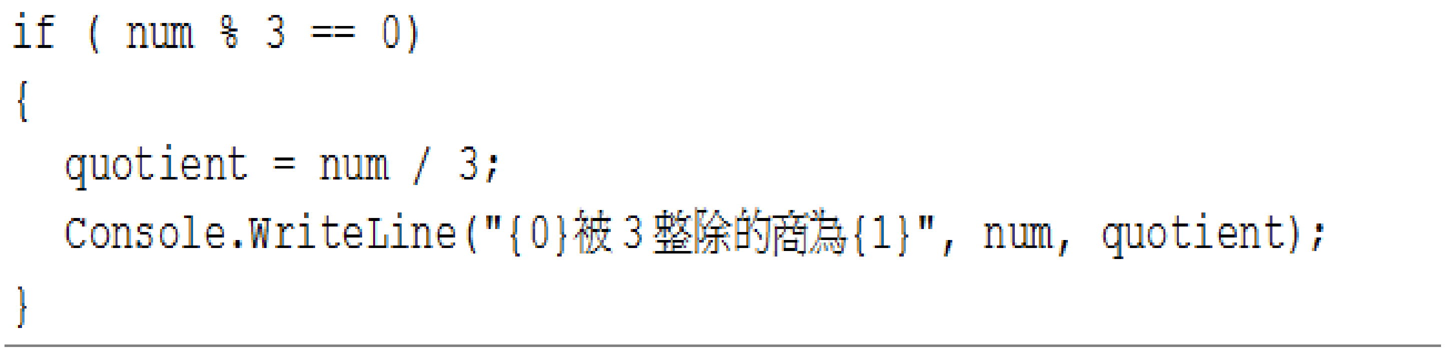
{} can be leave out if the statement is only in 1 line, usage:

```
if ( [condition] )
    [statement]
```

Ex1: get the absolute value of the number “num”: usage:

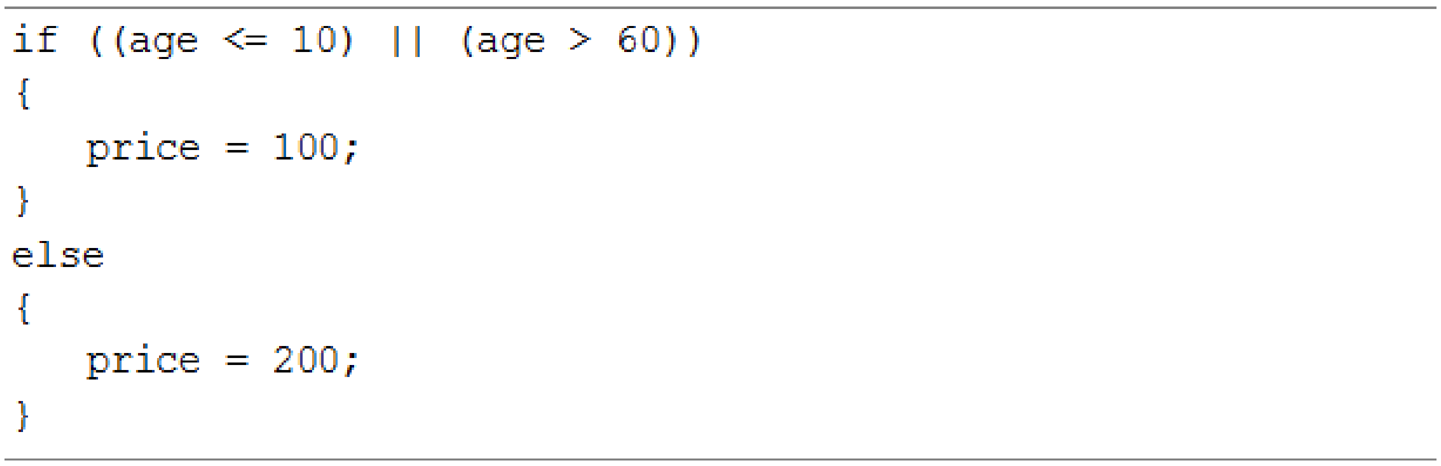
```
if ( num < 0)
   num = -num;
```

Ex2: if “num” is a multiple of 3, show the quotient of “num” which is divided by 3. Usage:



```
if ( num % 3 == 0)
{
   quotient = num / 3;
   Console.WriteLine("{0}被3整除的商為{1}", num, quotient);
}
```

Ex3: the price is 100 dollars if the age is <= 10 or > 60, otherwise, the price is 200 dollars. Usage:



```
if ((age <= 10) || (age > 60))
{
    price = 100;
}
else
{
    price = 200;
}
```

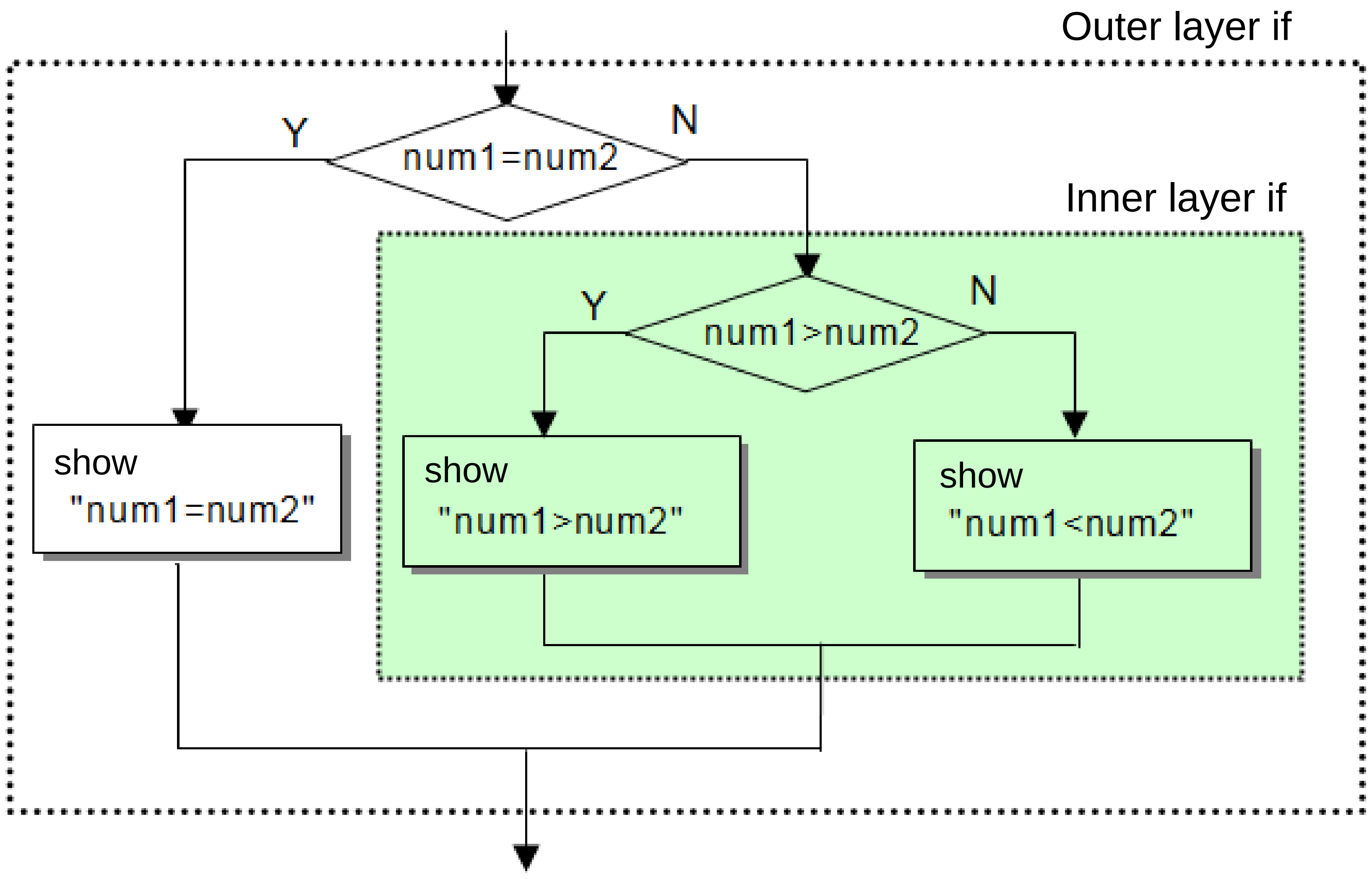
Nested if

- Nested if is formed by the if-else section which has another if-else section inside
- 3 conditions to determine
  complete 2 if-else statements to form nested if

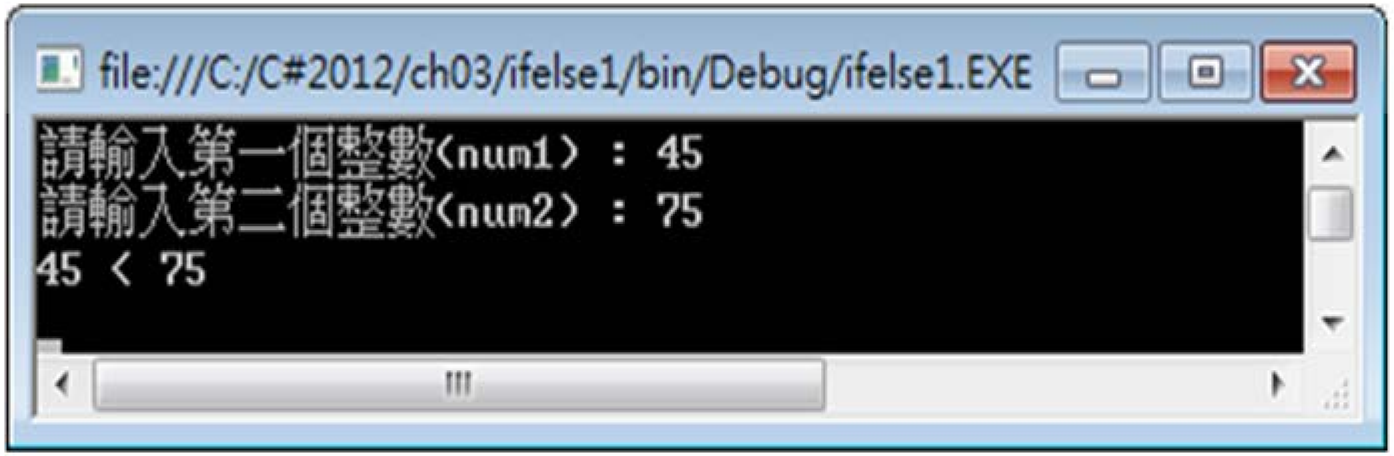
Practice(ifelse1):

Try to write a program to get integer input from keyboard, then:

1. If num1 = num2, show “num1 = num2”
2. If num1 > num2, show “num1 > num2”
3. If num1 < num2, show “num1 < num2”


Outer layer if
Y
num1=num2
N
Inner layer if
Y
num1>num2
N
show "num1=num2"
show "num1>num2"
show "num1<num2"

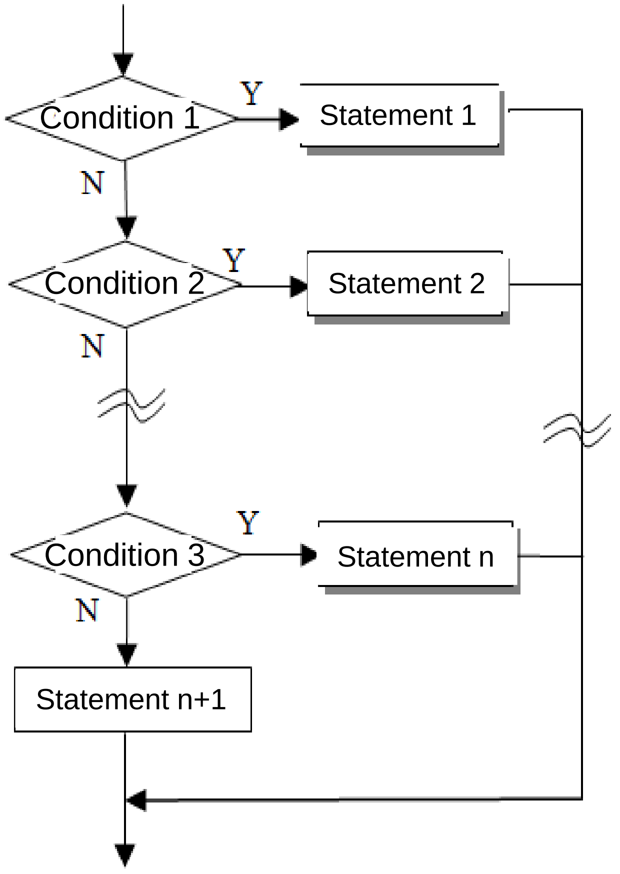
Practice(ifelse1):



```
請輸入第一個整數(num1) : 45
請輸入第二個整數(num2) : 75
45 < 75
```

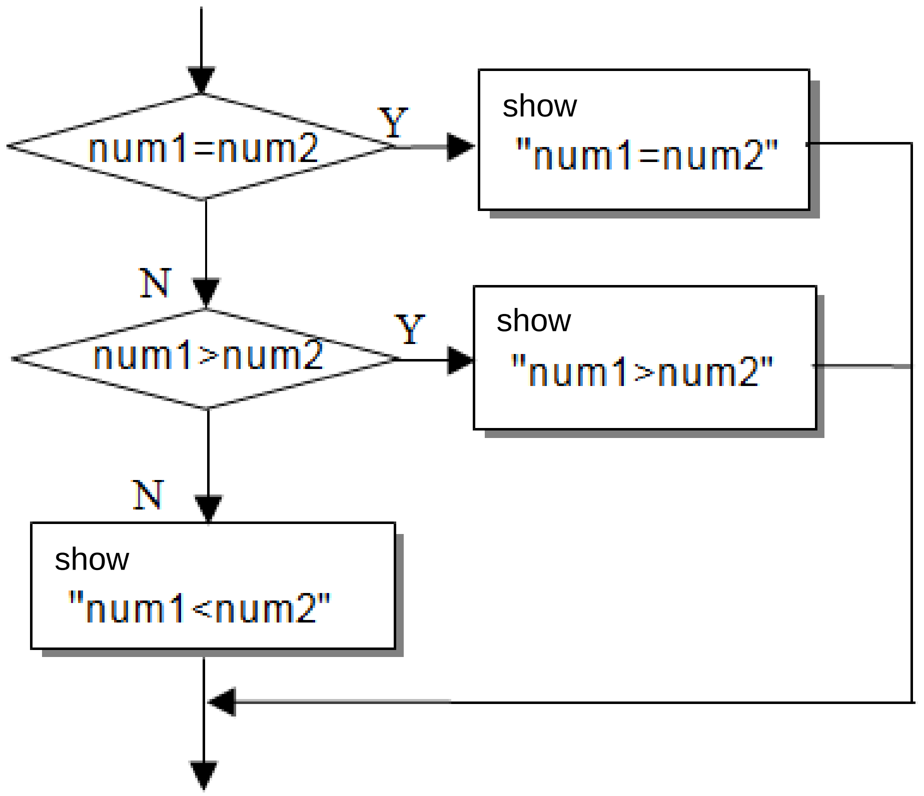
3.1.2 if…else if…else Multiple Selection

```
if ( [condition 1] )
{
    [statement 1]
}
else if ( [condition 2] )
{
    [statement 2]
}
    …
else if ( [condition n] )
{
    [statement n]
}
else
{
    [statement n+1]
}
```



Practice(ifelseif1):

From the former practice, use if…else if…else multiple selection statements to rewrite the program



3.1.3 switch Multiple Selection

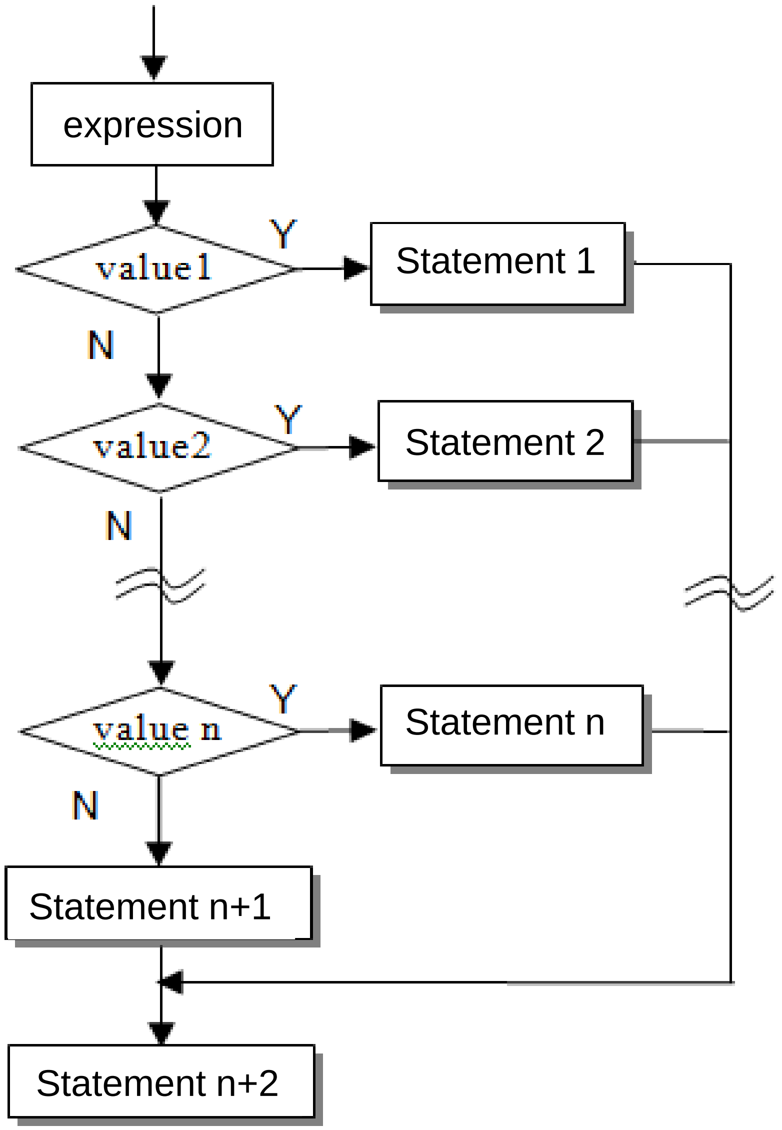
- **Difference**

  ① if … else if … else can use many different conditions

  ② switch allows only 1 statement

- **Too many "if" statements cause complexity and low maintainability, but "switch" statement does not**

```
switch ( [expression] )
{
    case [value 1]:
        [statement 1]
        break;
    case [value 2]:
        [statement 2]
        break;
    …
    case [value n]:
        [statement n]
        break;
    default:
        [statement n+1];
}
[statement n+2];
```



Case in different style

① If condition 1, 2, 4 is true:

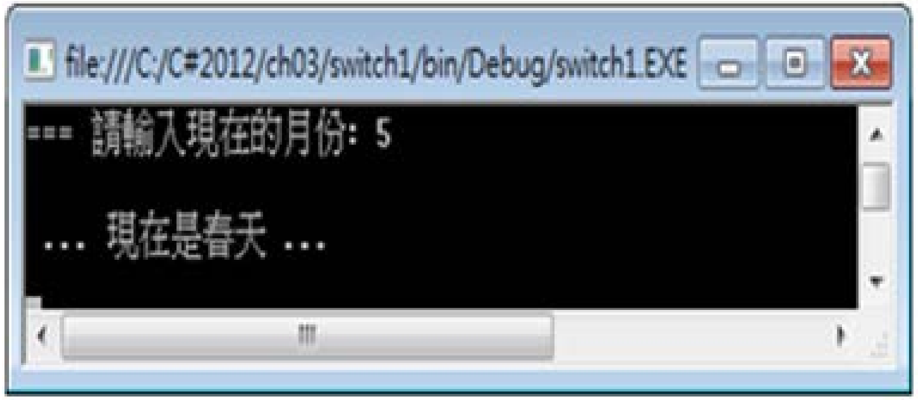
```
case 1 :
case 2:
case 4:

        Statement;

break;
```

② If result of condition is “Y” or “y” or true:

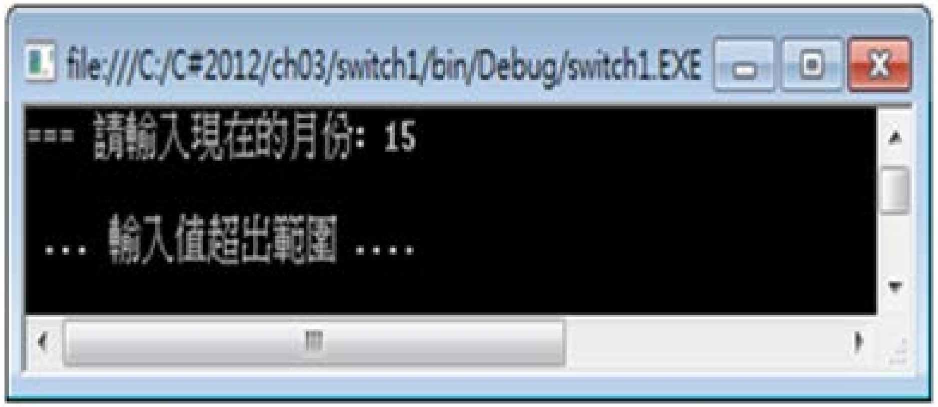
```
case "y" :
case "Y" :

        Statement;

break;
```

Practice(switch1):

Try to use switch statement to get month input from keyboard and show the season of the month. If the input is not 1~12, show message.



```
=== 請輸入現在的月份: 5

 ... 現在是春天 ...
```



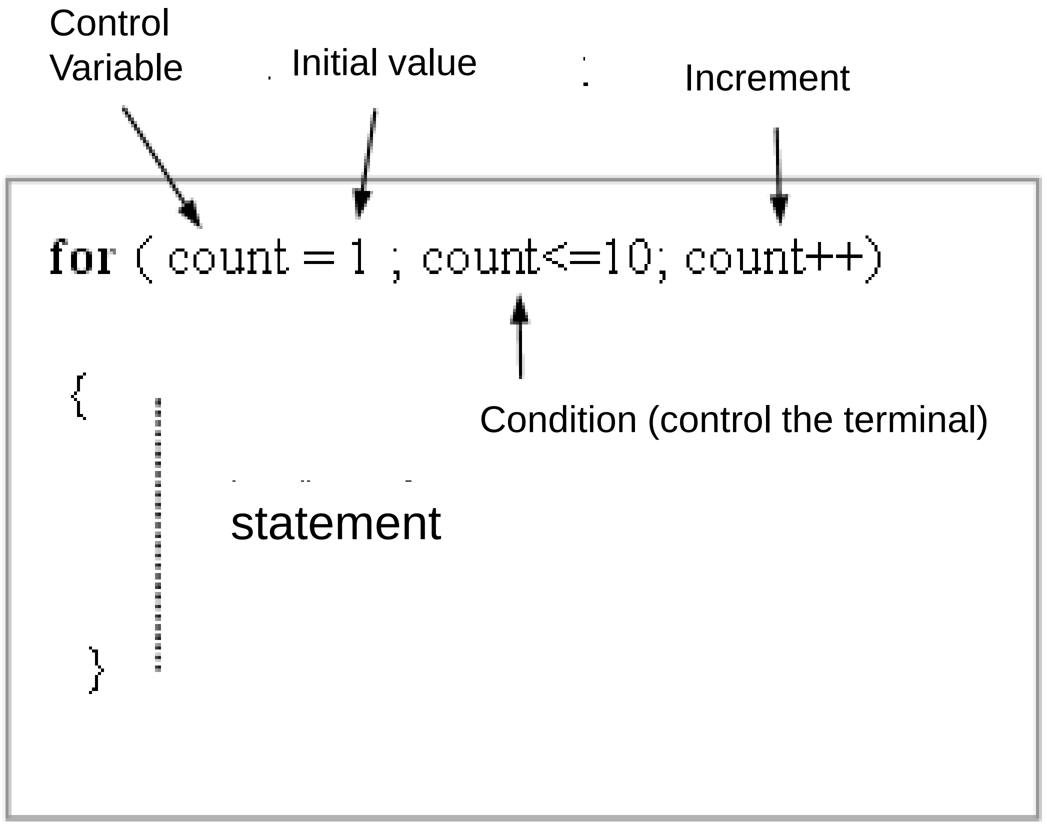
```
=== 請輸入現在的月份: 15

 ... 輸入值超出範圍 ....
```

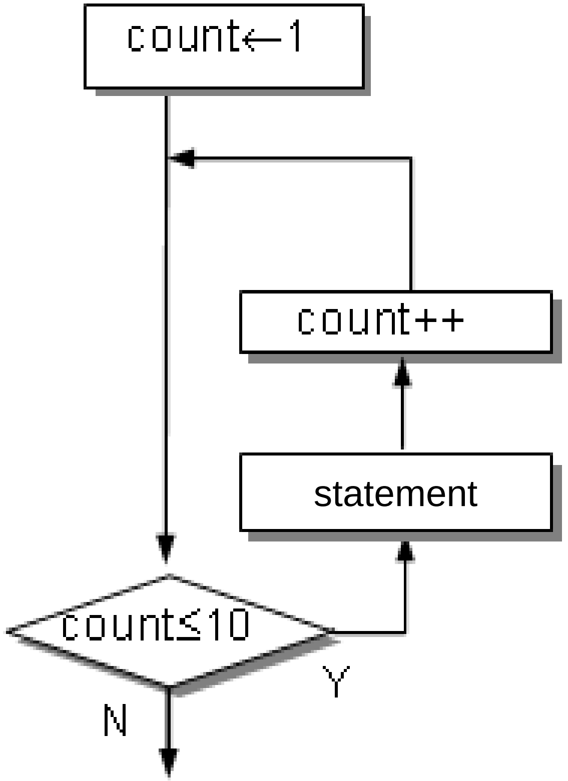
3.2 Iteration Statements

- Also called loop
- Some sections in the program have to be repeated in several times or until the condition is not fulfilled
- “for” statement: assign the number of times, called counter controlled loop
- “while” statement: run according to the condition, called condition controlled loop

3.2.1 for

- **Counter controlled loop begins from left bracket of for loop and ends at right bracket**


Control Variable
Initial value
Increment
for (count = 1 ; count<=10; count++)
{
statement
}
Condition (control the terminal)



- Use break statement to leave from halfway for loop
- Use continue statement to jump to the beginning of "for" loop immediately and carry on execution

Ordinary for loop usage:

① for (k=1 ;   k<= 5 ; k++)

k = 1, 2, 3, 4, 5. The loop executes 5 times

② for (k=1 ; k<=5 ; k+=2)

k = 1, 3, 5. The loop executes 3 times

③ **Initial value and iterator can be a decimal**

for (k=-0.5 ; k<=1.5 ; k+= 0.5)

k = -0.5, 0, 0.5, 1.0, 1.5. The loop executes 5 times

④ **Iterator is decrement**

for (k=6 ; k>=1 ; k-=2)

k = 6, 4, 2. The loop executes 3 times

⑤ **2 or more initial values, separate them by comma(,):**

for (x=1, y=5 ; x<3 && y>2 ;x++ , y--)

x=1 & y=5; x=2 & y=4; the loop executes 2 times

⑥ **Initial values and condition can have expressions**

for (k=x ; k<=y+9 ; k+=2)

if x=1, y=-2, then k = 1,3,5,7. The loop executes 4 times

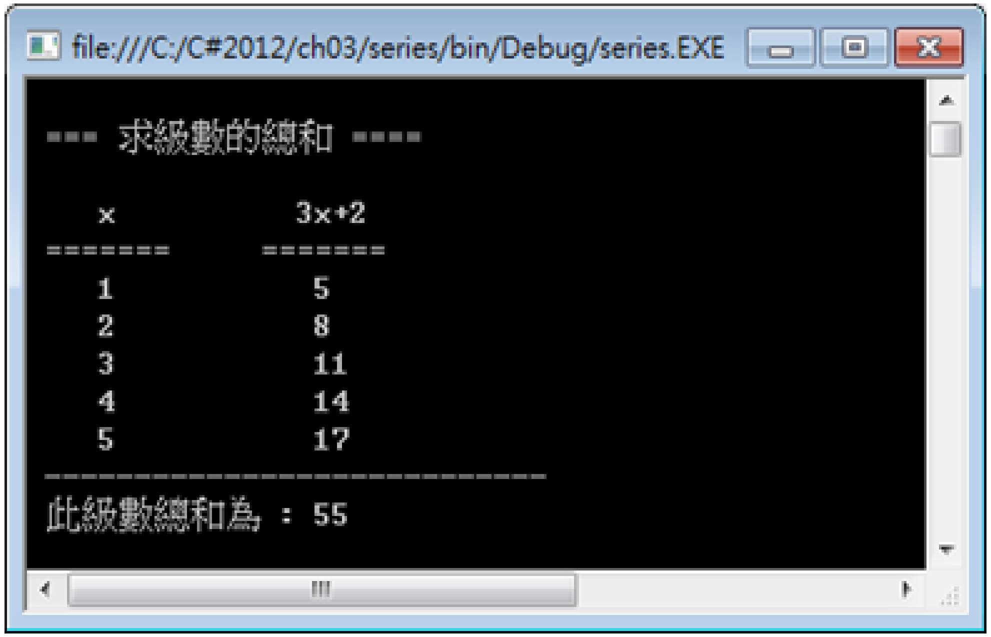
⑦ **Infinite loop**

for (; ;)

Practice(series):

Get the sum of the following series

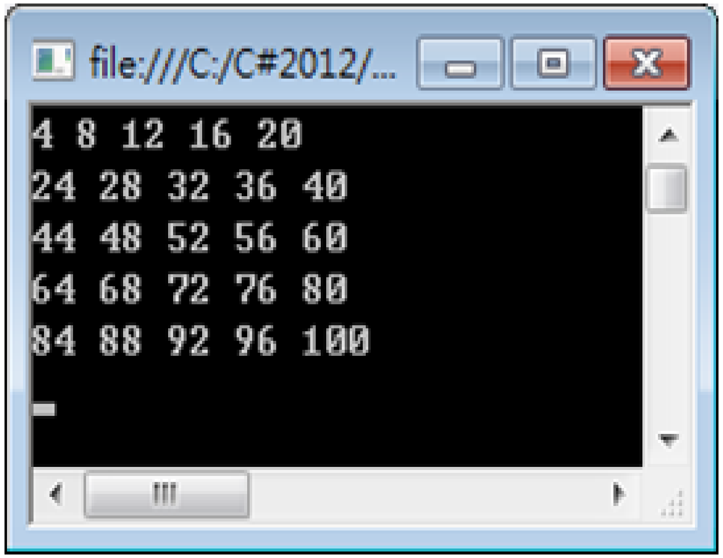
$$\sum_{x=1}^{5}(3x+2) = \underset{x=1}{\underline{5}} + \underset{x=2}{\underline{8}} + \underset{x=3}{\underline{11}} + \underset{x=4}{\underline{14}} + \underset{x=5}{\underline{17}} = ?$$



```
=== 求級數的總和 ====

   x          3x+2
=======     =======
   1           5
   2           8
   3           11
   4           14
   5           17
-----------------------------
此級數總和為 : 55
```

Practice(for1):

Try to write a program to show the multiples of 4 between 1 and 100.
Print 5 numbers in 1 line

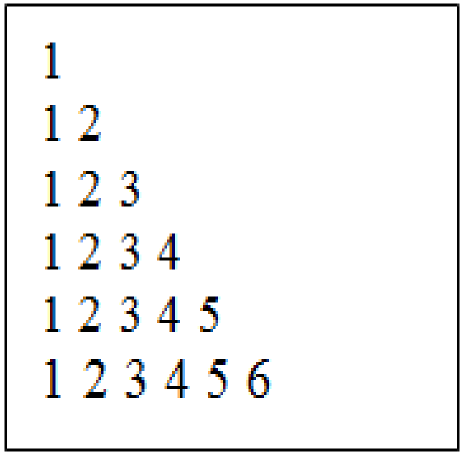


```
4 8 12 16 20
24 28 32 36 40
44 48 52 56 60
64 68 72 76 80
84 88 92 96 100
```

3.2.2 Nested Loop

- **A loop which has loops inside forms a nested loop usually used in 2-d array**
- **Use nested loop to show numbers like a ladder**
  1st stair shows 1,
  2nd stair shows 1 2,
  3rd stair shows 1 2 3 …
  a space between numbers, show 6 stairs

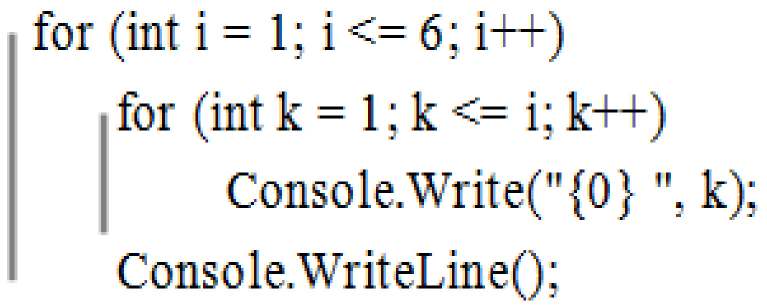
Practice(forsample):
Assume i is the number of llines, k stands for the number to show



```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

When i = 1, show k = 1~1
When i = 2, show k = 1~2
…
When i = 6, show k = 1~6

Source code:

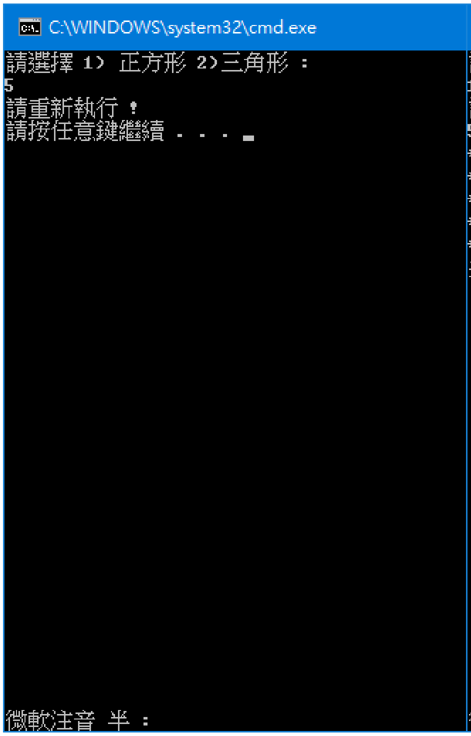


```
for (int i = 1; i <= 6; i++)
    for (int k = 1; k <= i; k++)
        Console.Write("{0} ", k);
    Console.WriteLine();
```

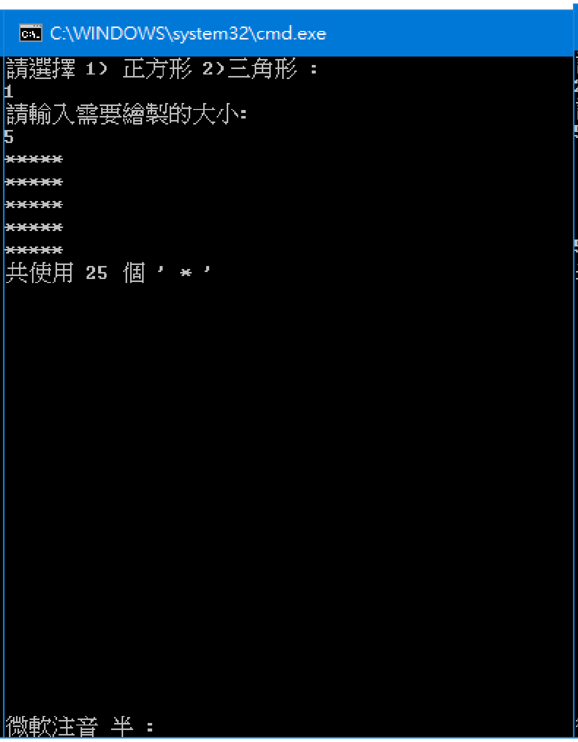
Practice 2

- Tips:
  - use for loop to draw a triangle and a square.
  - Use ‘*’ to draw square
  - Use ‘number’ to draw a triangle
  - When you draw triangle, think about what you need to draw first, ‘space’ or ‘number’.

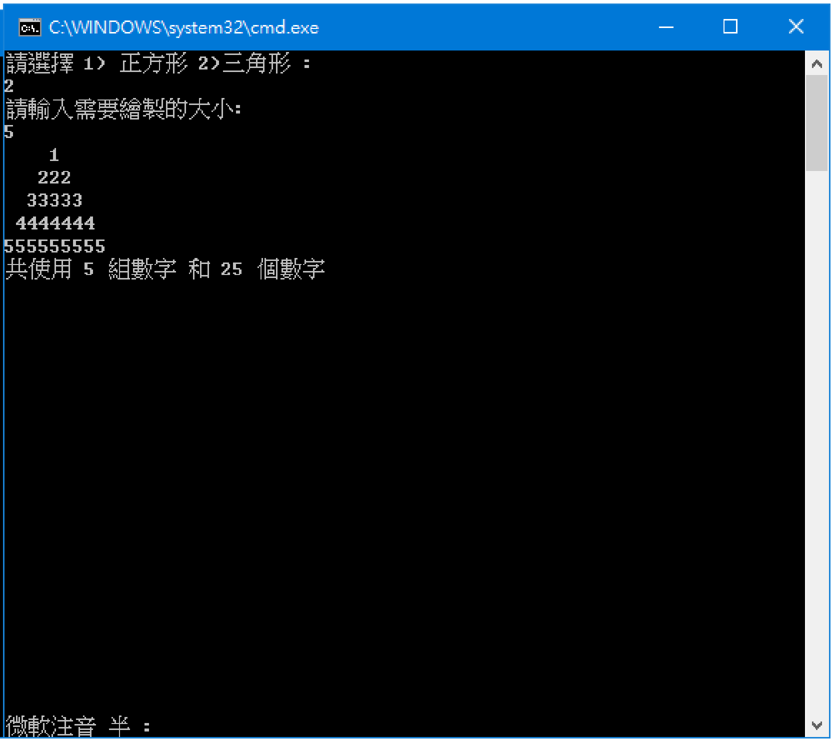
Practice 2



```
C:\WINDOWS\system32\cmd.exe
請選擇 1> 正方形 2>三角形 :
5
請重新執行 !
請按任意鍵繼續 . . .

微軟注音 半 :
```



```
C:\WINDOWS\system32\cmd.exe
請選擇 1> 正方形 2>三角形 :
1
請輸入需要繪製的大小:
5
*****
*****
*****
*****
*****
共使用 25 個 ' * '

微軟注音 半 :
```



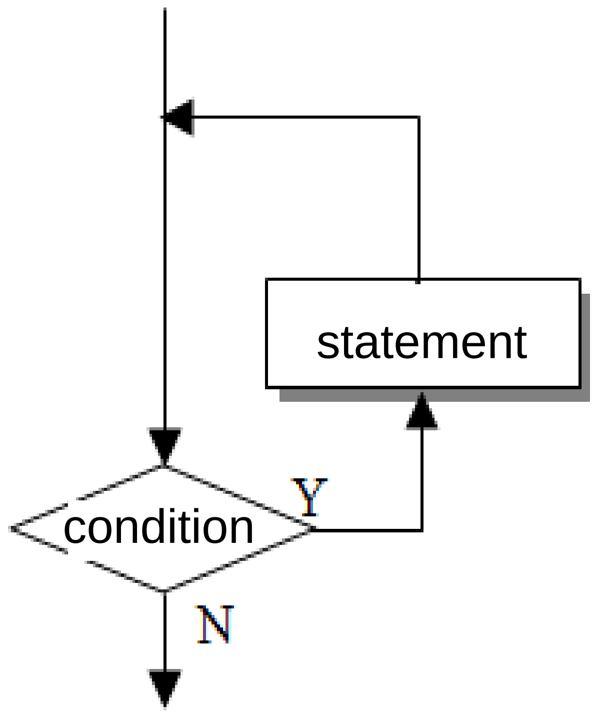
```
C:\WINDOWS\system32\cmd.exe
請選擇 1> 正方形 2>三角形 :
2
請輸入需要繪製的大小:
5
    1
   222
  33333
 4444444
555555555
共使用 5 組數字 和 25 個數字

微軟注音 半 :
```

3.2.3 Pre-test Loop

- **The condition statement is at the beginning of the loop**
- **Decide to enter the loop or not by the result of condition**
  ① fulfilled, execute the statements in the loop once and back to the beginning of the loop
  ② not fulfilled, exit the loop
- **First time enter the loop, and the condition is false, exit the loop immediately**

3.2.3 Pre-test Loop

Grammar	Flow chart

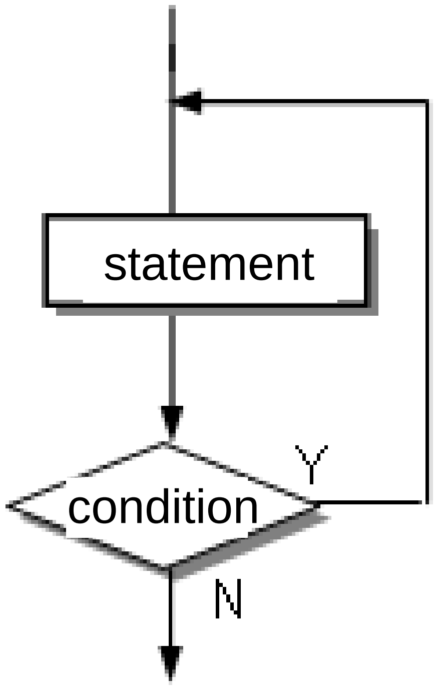
```
while ( [condition] )
{
    [statement]
}
```
	

3.2.4 Post-test Loop

- The condition statement is put at the end of the loop
- First run does not examine the condition, then the condition is checked at the end of loop
  ① the statements run one more time if the condition is fulfilled, then the condition at the end of loop is examined again
  ② exit the loop until the condition is not fulfilled
- The statement in the loop runs at least one time

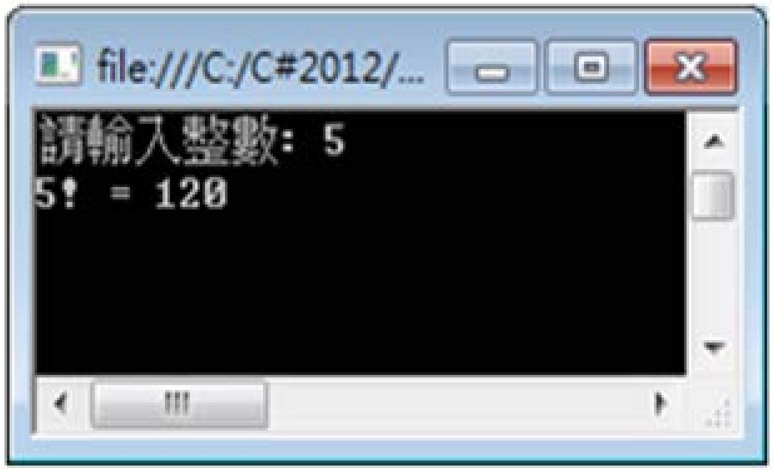
3.2.4 Post-test Loop

Grammar	Flow chart

do
{
    [statement]
} while ([condition]);
	

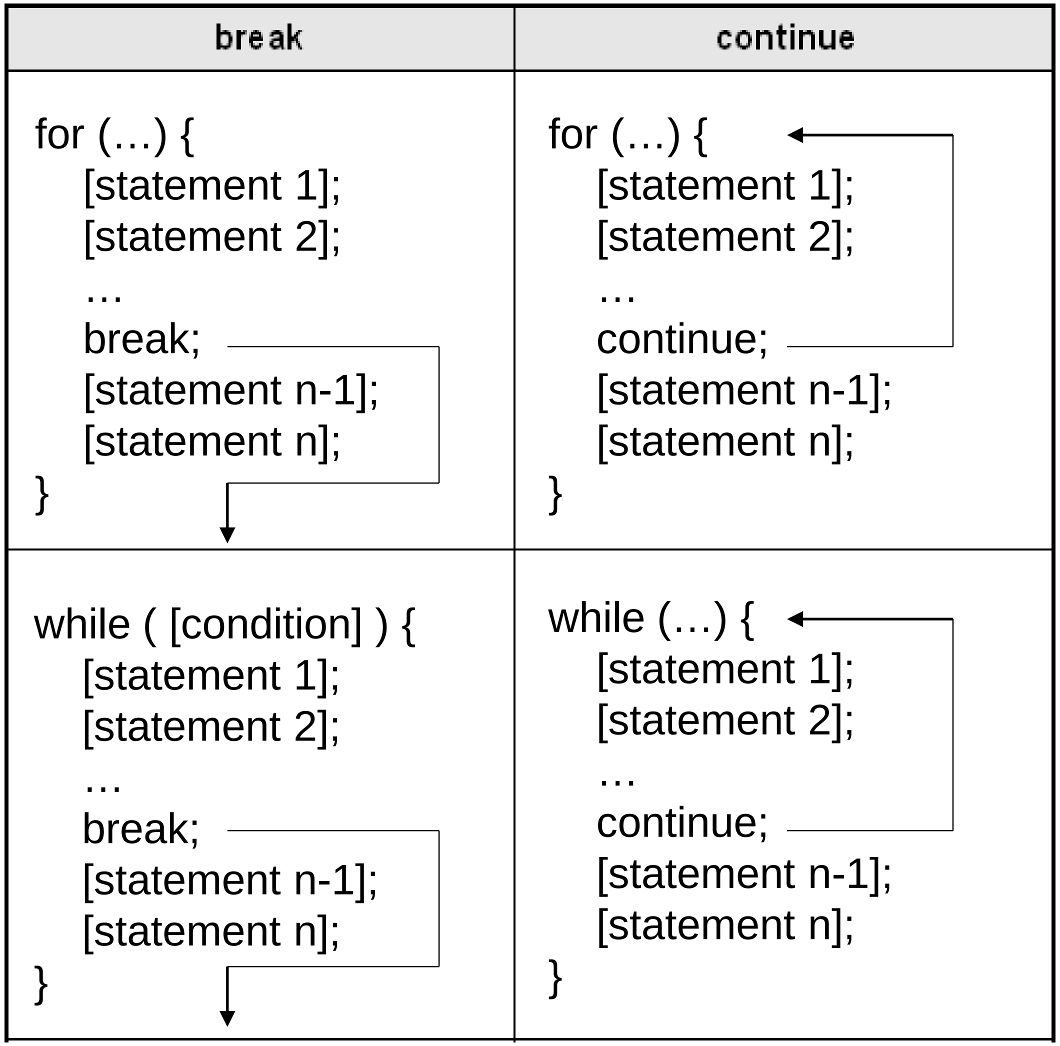
Practice(factorial):

Try to write a program which uses pre-test loop to calculate factorial. First the user inputs an integer, then the factorial of the number is computed.



```
請輸入整數: 5
5! = 120
```

3.3 break and continue



break

```
for (…) {
    [statement 1];
    [statement 2];
    …
    break;
    [statement n-1];
    [statement n];
}
```

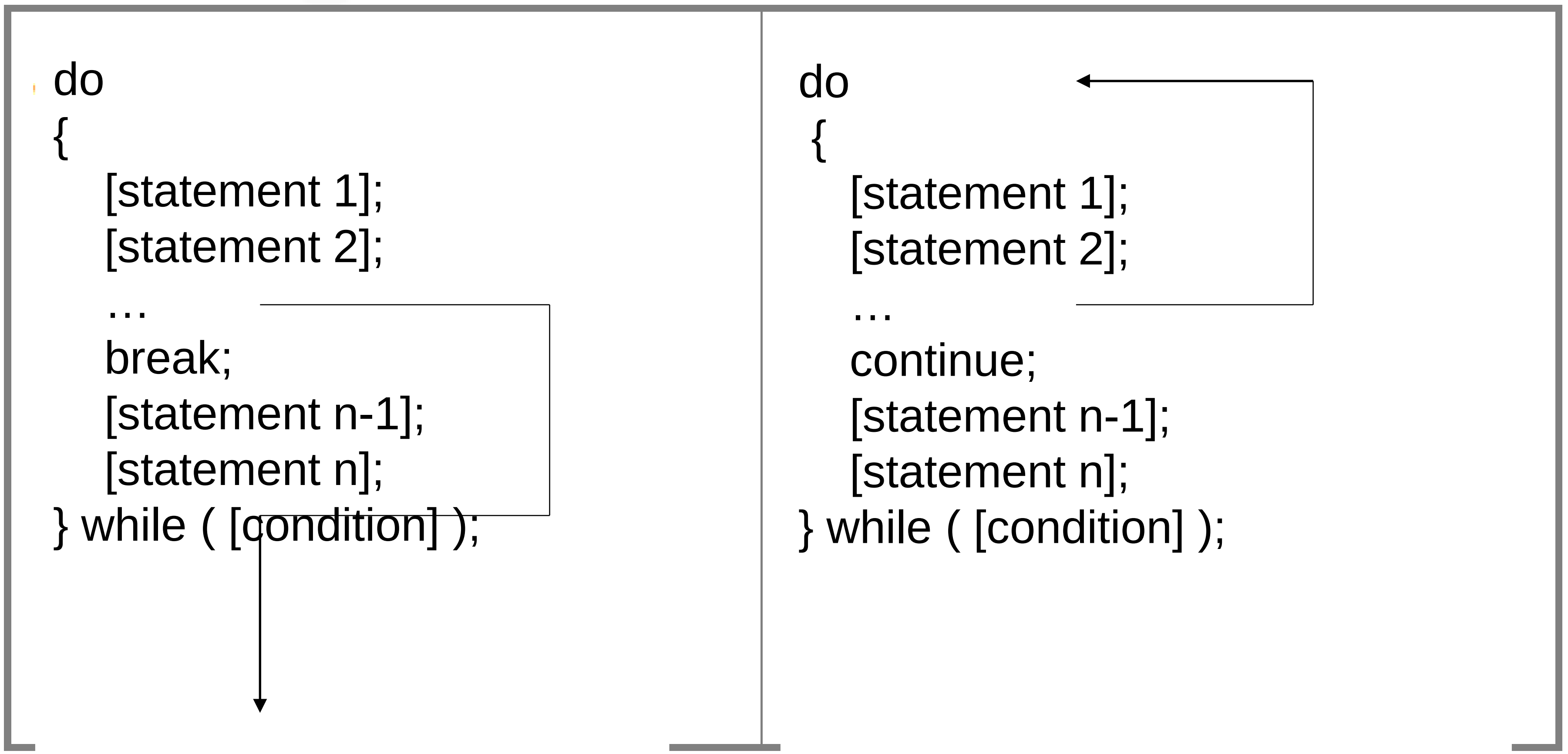
```
while ( [condition] ) {
    [statement 1];
    [statement 2];
    …
    break;
    [statement n-1];
    [statement n];
}
```

continue

```
for (…) {
    [statement 1];
    [statement 2];
    …
    continue;
    [statement n-1];
    [statement n];
}
```

```
while (…) {
    [statement 1];
    [statement 2];
    …
    continue;
    [statement n-1];
    [statement n];
}
```

3.3 break and continue

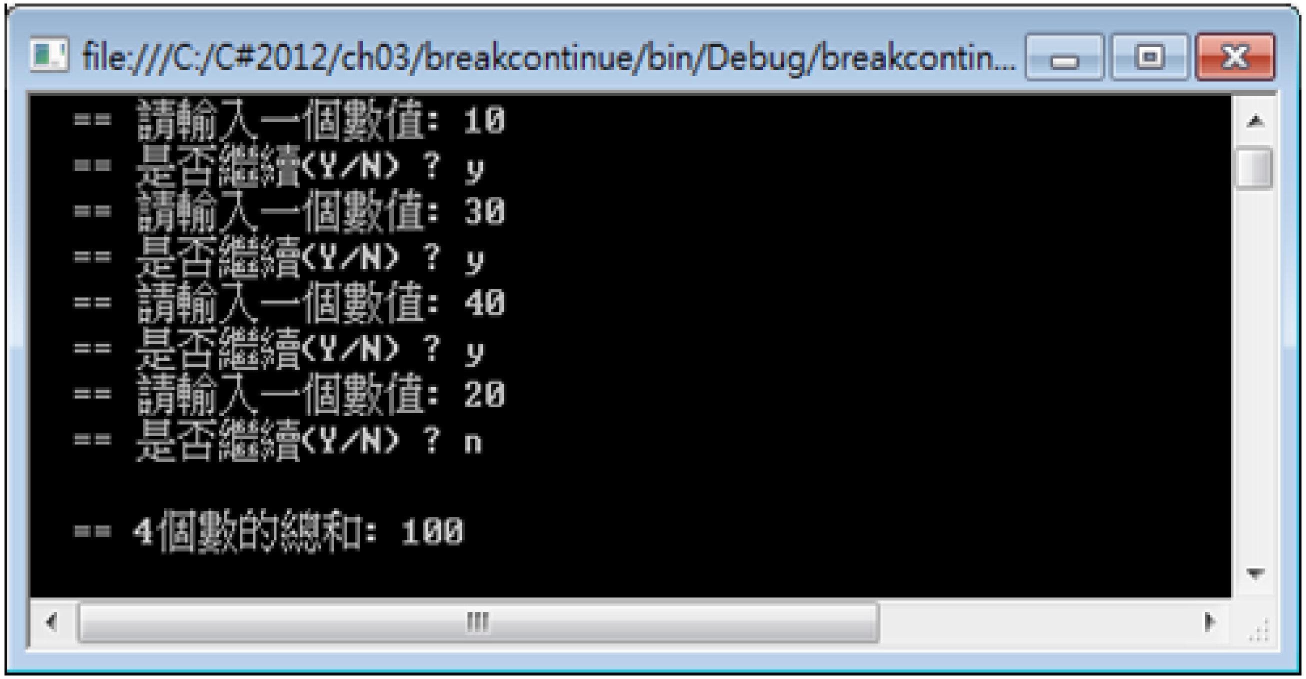


```
do
{
    [statement 1];
    [statement 2];
    …
    break;
    [statement n-1];
    [statement n];
} while ( [condition] );
```

```
do
 {
    [statement 1];
    [statement 2];
    …
    continue;
    [statement n-1];
    [statement n];
} while ( [condition] );
```

Practice(breakcontinue):

Try to write a program which accumulate numbers, use break and continue to decide whether continue accumulating or not in do…while loop.



file:///C:/C#2012/ch03/breakcontinue/bin/Debug/breakcontin...

```
 == 請輸入一個數值: 10
 == 是否繼續(Y/N) ? y
 == 請輸入一個數值: 30
 == 是否繼續(Y/N) ? y
 == 請輸入一個數值: 40
 == 是否繼續(Y/N) ? y
 == 請輸入一個數值: 20
 == 是否繼續(Y/N) ? n

 == 4個數的總和: 100
```

3.4 Program Debug

- **The unexpected result may cause**
  ① compilation error
  ② runtime error
- **Syntax error**
  the error occurs during compilation. The identifier is marked by wavy blue underline and unrecognizable.

3.4 Program Debug Continue…

- Logic Error
  no error occurs after compilation completed, but the expected result does not happen when the program is running
- Logic error is not grammatical error
  - program flow
  - statement
  - wrong variable application

3.5 Exception

- Error occurs when the program is running
- C# provides a structured and easy-to-control solution to handle the unexpected condition

```
try
{
    [try statement]
}
catch (exception1 ex)
{
    [catch statement]
}
catch (exception2 ex)
{
    [catch statement]
}
    …
finally
{
    [finally statement]
}
```

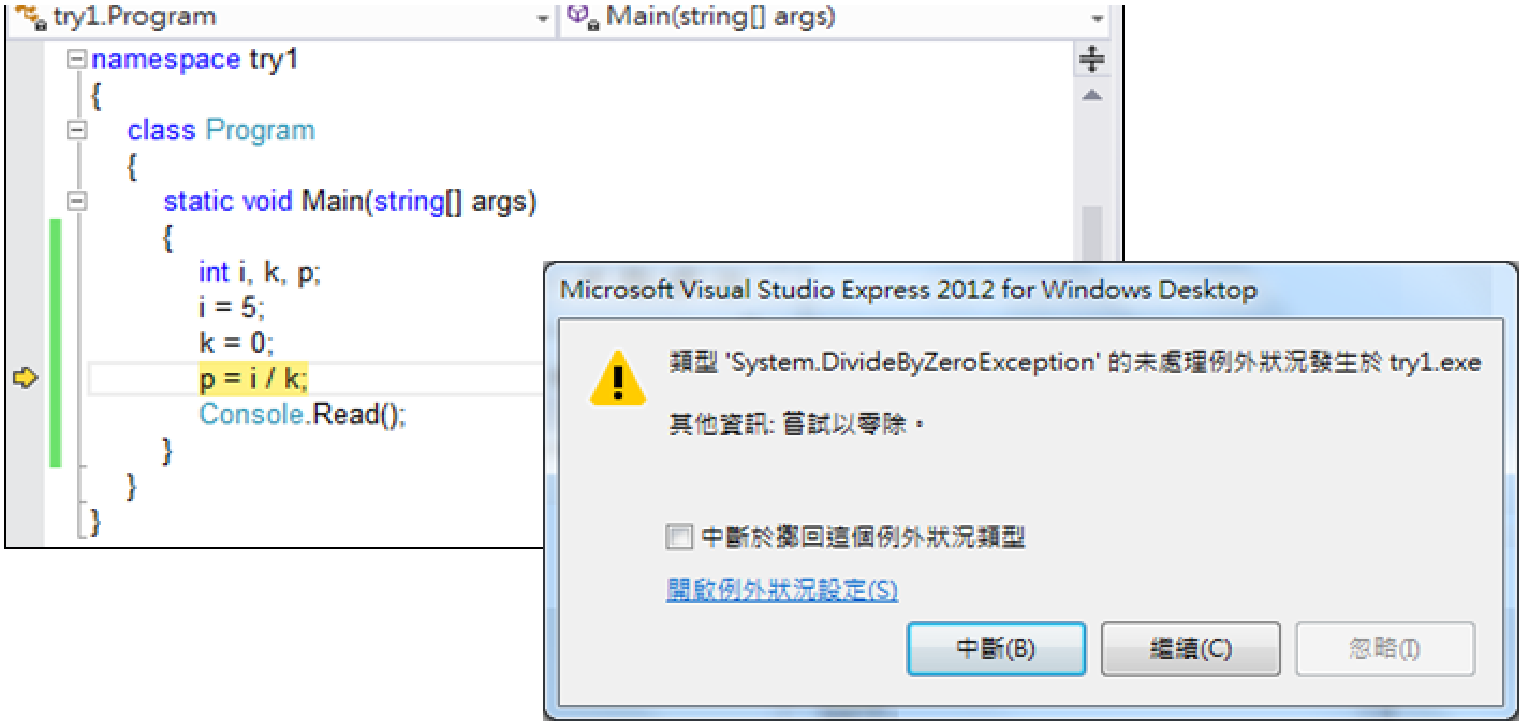
3.5 Exception

Continue…

Exception classes	Error reason
ArgumentOutOfRangeException	Argument’s data type is out of the range defined by the function parameter
DivideByZeroException	Divisor is zero
IndexOutOfRangeException	Array index is out of the maximum size
InvalidCastException	Data type conversion error
OverFlowException	Data over flow
Exception	Runtime error

Practice(try1):

Try to write a program which can cause DivideByZeroException. First i, k, p are declared. The initial value of i is 5, k is 0. The program is terminated when i/k causes DivdeByZeroException.



```
namespace try1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, k, p;
            i = 5;
            k = 0;
            p = i / k;
            Console.Read();
        }
    }
}
```

Microsoft Visual Studio Express 2012 for Windows Desktop

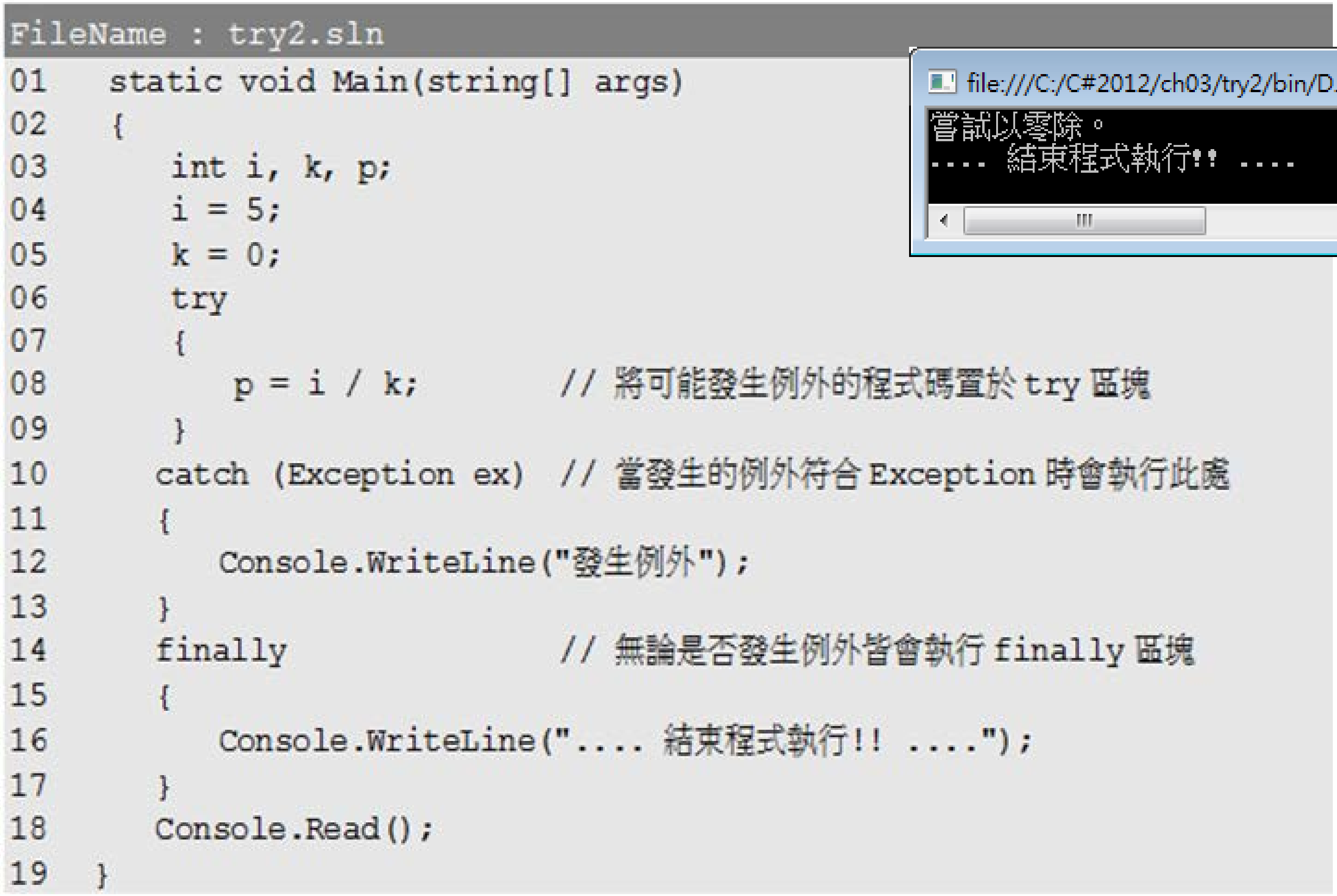
類型 'System.DivideByZeroException' 的未處理例外狀況發生於 try1.exe

其他資訊: 嘗試以零除。

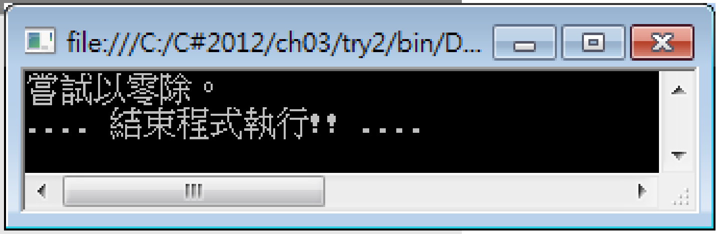
中斷於擲回這個例外狀況類型

Practice(try2):

From the former practice, try to insert try…catch to handle the exception.



```
FileName : try2.sln
static void Main(string[] args)
{
    int i, k, p;
    i = 5;
    k = 0;
    try
    {
        p = i / k;          // 將可能發生例外的程式碼置於 try 區塊
    }
   catch (Exception ex)   // 當發生的例外符合 Exception 時會執行此處
   {
       Console.WriteLine("發生例外");
   }
   finally                // 無論是否發生例外皆會執行 finally 區塊
   {
       Console.WriteLine(".... 結束程式執行!! ....");
   }
   Console.Read();
}
```

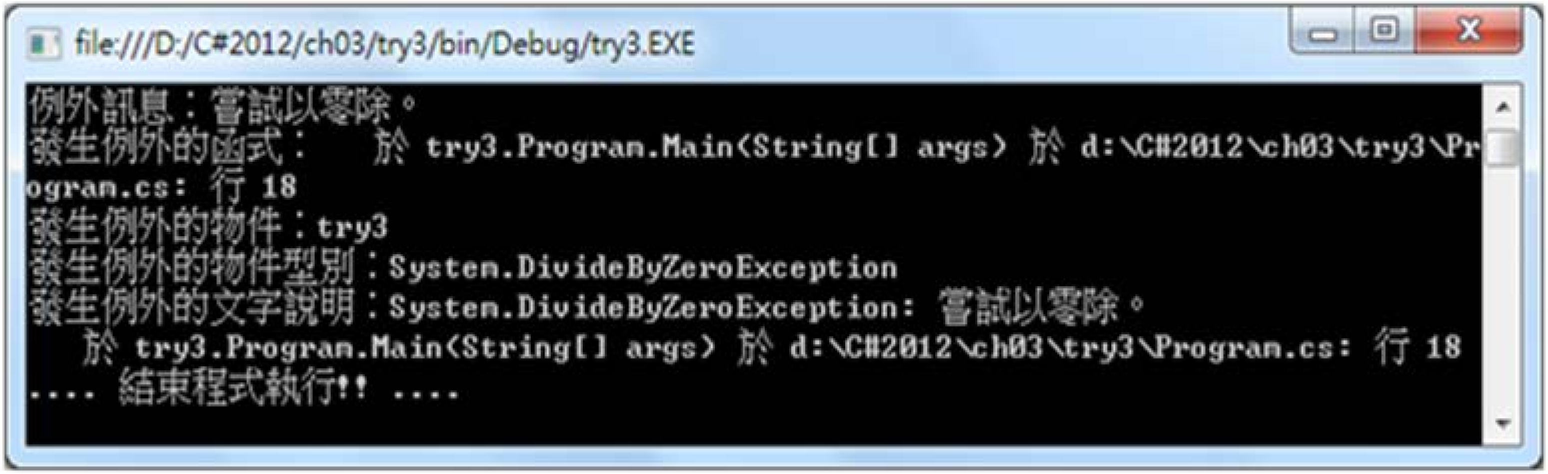


file:///C:/C#2012/ch03/try2/bin/D...

```
嘗試以零除。
.... 結束程式執行!! ....
```

Attributes and methods in common use

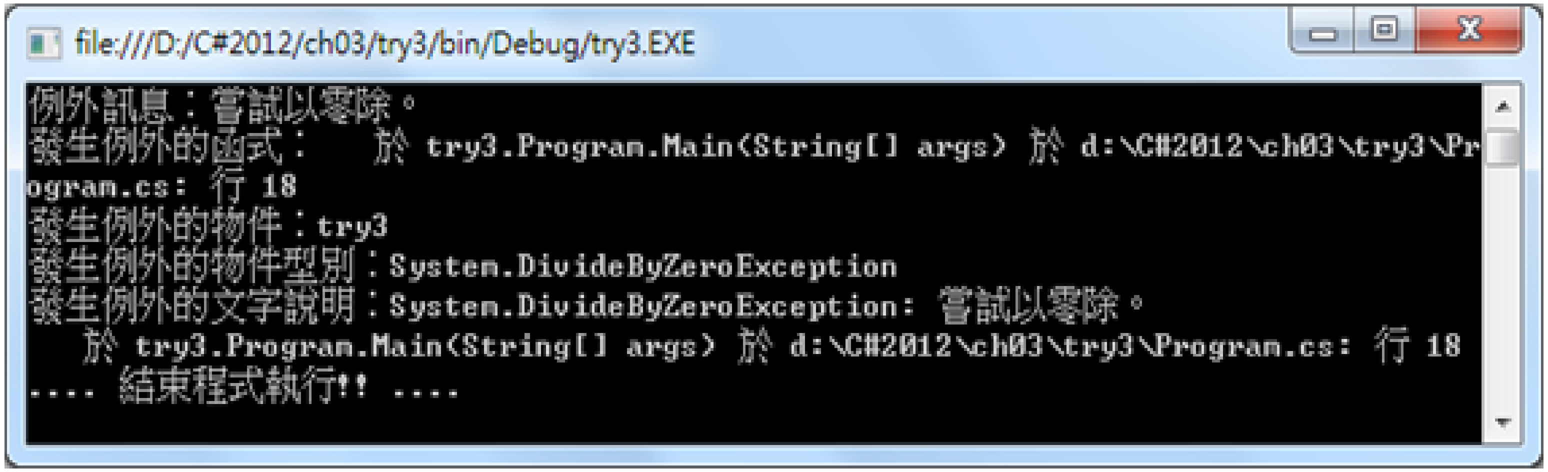
Members of exception	Description
GetType	Get data type of exception object
ToString	Get text description of exception object
Message	Get exception message
Source	Get application or object which cause exception
StackTrace	Get methods or functions which cause exception



```
例外訊息：嘗試以零除。
發生例外的函式：   於 try3.Program.Main(String[] args) 於 d:\C#2012\ch03\try3\Pr
ogram.cs: 行 18
發生例外的物件：try3
發生例外的物件型別：System.DivideByZeroException
發生例外的文字說明：System.DivideByZeroException: 嘗試以零除。
   於 try3.Program.Main(String[] args) 於 d:\C#2012\ch03\try3\Program.cs: 行 18
.... 結束程式執行!! ....
```

Practice(try3):

From the former practice, please use GetType, ToString, Message, Source, StackTrace members of exception object to show the information of exception



```
例外訊息：嘗試以零除。
發生例外的函式：   於 try3.Program.Main(String[] args) 於 d:\C#2012\ch03\try3\Pr
ogram.cs: 行 18
發生例外的物件：try3
發生例外的物件型別：System.DivideByZeroException
發生例外的文字說明：System.DivideByZeroException: 嘗試以零除。
   於 try3.Program.Main(String[] args) 於 d:\C#2012\ch03\try3\Program.cs: 行 18
.... 結束程式執行!! ....
```

FileName : try3.sln

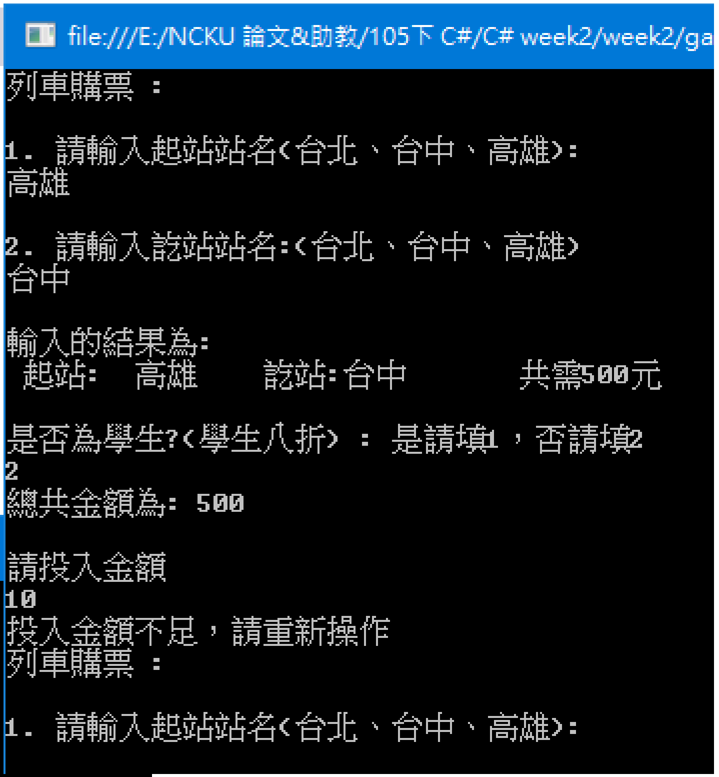
```
static void Main(string[] args)
{
    int i, k, p;
    i = 5;
    k = 0;
    try
    {
        p = i / k; // 將可能發生例外的程式碼置於 try 區塊
    }
    catch (DivideByZeroException ex)
    {
        Console.WriteLine("例外訊息：{0}", ex.Message);
        Console.WriteLine("發生例外的函式：{0}", ex.StackTrace);
        Console.WriteLine("發生例外的物件：{0}", ex.Source);
        Console.WriteLine("發生例外的物件型別：{0}", ex.GetType());
        Console.WriteLine("發生例外的文字說明：{0}", ex.ToString());
    }
    finally          // 無論是否發生例外，皆會執行 finally 區塊中的程式碼
    {
        Console.WriteLine(".... 結束程式執行!! ....");
    }
    Console.Read();
}
```

Ticket machine

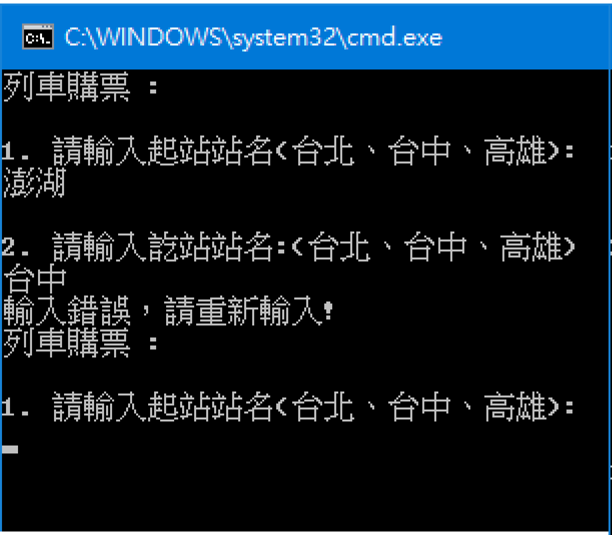
- Tips:
  - Use while loop to run the program continually.
  - Use if to decide how much price you choose from the station.
  - Use if to determine if the program will keep running or not.
  - You have to enforce change the price into integer when you finish calculate the price of student.
  - Price: 台北<->台中 台中<->高雄 500

    台北<->高雄 1000

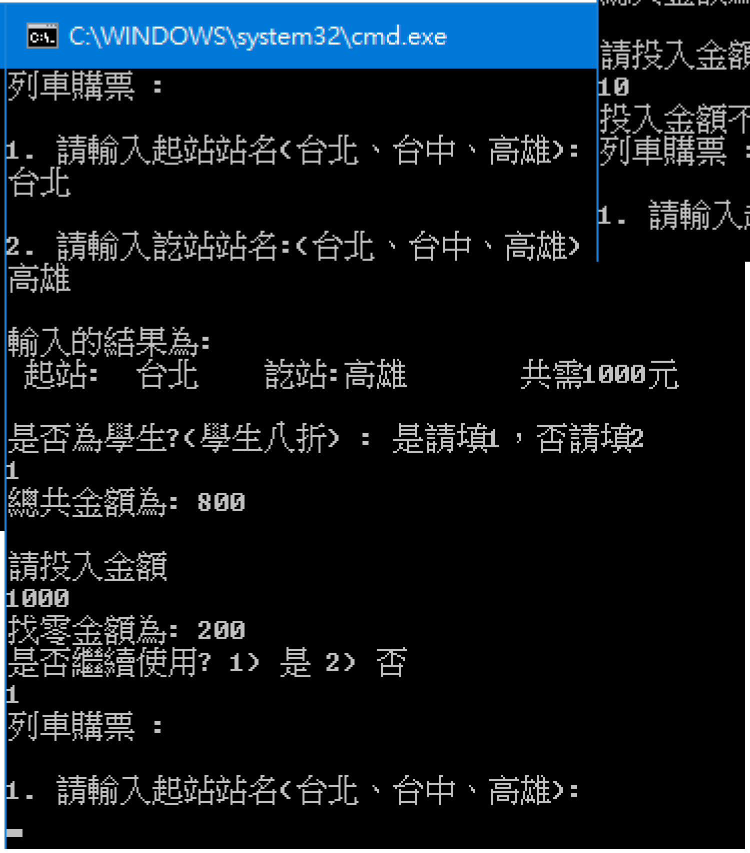
Ticket machine



```
列車購票 :

1. 請輸入起站站名(台北、台中、高雄):
高雄

2. 請輸入訖站站名:(台北、台中、高雄)
台中

輸入的結果為:
 起站:  高雄     訖站:台中          共需500元

是否為學生?(學生八折) : 是請填1 ，否請填2
2
總共金額為: 500

請投入金額
10
投入金額不足，請重新操作
列車購票 :

1. 請輸入起站站名(台北、台中、高雄):
```



```
列車購票 :

1. 請輸入起站站名(台北、台中、高雄):
澎湖

2. 請輸入訖站站名:(台北、台中、高雄)
台中
輸入錯誤，請重新輸入!
列車購票 :

1. 請輸入起站站名(台北、台中、高雄):
_
```



```
列車購票 :

1. 請輸入起站站名(台北、台中、高雄):
台北

2. 請輸入訖站站名:(台北、台中、高雄)
高雄

輸入的結果為:
 起站:  台北     訖站:高雄          共需1000元

是否為學生?(學生八折) : 是請填1 ，否請填2
1
總共金額為: 800

請投入金額
1000
找零金額為: 200
是否繼續使用? 1) 是 2) 否
1
列車購票 :

1. 請輸入起站站名(台北、台中、高雄):
_
```

The End

Take a Break …..