

## บทที่ 4

### การทำซ้ำ

#### ◆ ความหมายของการทำซ้ำ

การทำซ้ำหรือการวนรอบ (loop) หมายถึง การกำหนดเงื่อนไขเพื่อให้โปรแกรมทำงานใน statement หรือ กลุ่มของ statement เดิมซ้ำๆ 1 ครั้งหรือมากกว่า โดยจำนวนครั้งของการทำงานจะเป็นไปตามเงื่อนไขที่กำหนดไว้ มีประโยชน์อย่างยิ่งทำให้การเขียนโปรแกรมยืดหยุ่นและสะดวก การเขียนโปรแกรมมีประสิทธิภาพด้วยคำสั่งกำหนดให้มีการทำซ้ำหรือวนรอบ

ใน C++ มี statement เพื่อสั่งให้โปรแกรมมีการทำงานแบบทำซ้ำ คือ **for**, **while** และ **do...** **while** ซึ่งจะกล่าวถึงรายละเอียดต่อไป

#### ◆ ลูป for

ลูป for จะเริ่มด้วยการนำค่าเริ่มต้นเปรียบเทียบกับเงื่อนไขที่กำหนดไว้ ถ้าเงื่อนไข **เป็นจริง** จะทำ statement ในลูป ถ้า **เป็นเท็จ** จะเลิกทำงานในลูป กรณีเมื่อตรวจสอบเงื่อนไขแล้วเป็นจริงเมื่อทำงานในลูปแล้ว ก็จะเพิ่มหรือลดค่าตัวแปรในเงื่อนไขอีก 1 หรือมากกว่าตามที่กำหนดไว้โดยอัตโนมัติ แล้วตรวจสอบเงื่อนไขอีกครั้ง ถ้าเงื่อนไข **เป็นเท็จ** จะเลิกทำซ้ำในลูป มีรูปแบบ statement ดังนี้

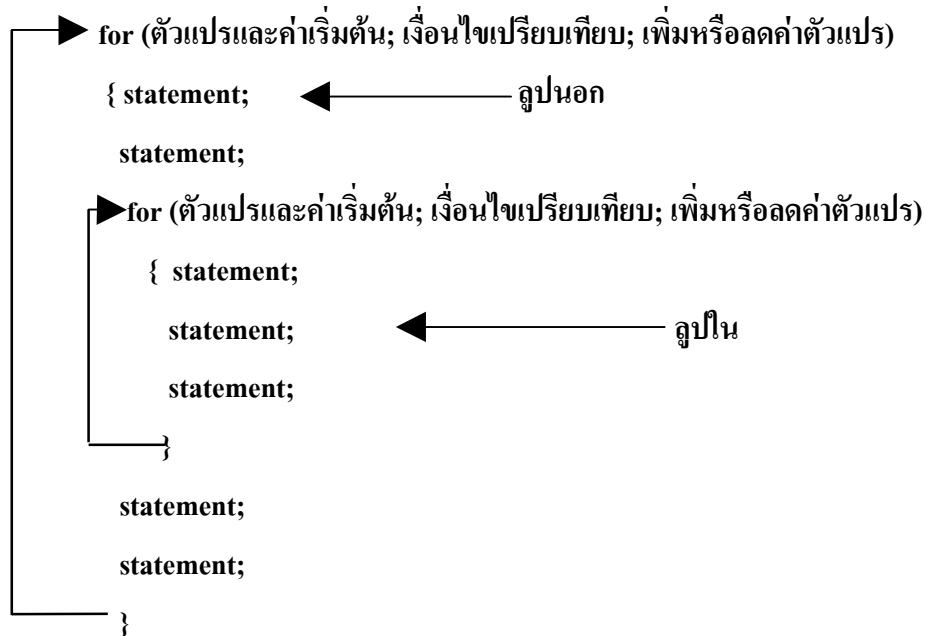
1. ลูป for ที่มี statement เดียว มีรูปแบบ ดังนี้

```
for (ตัวแปรและค่าเริ่มต้น; เงื่อนไขเปรียบเทียบ; เพิ่มหรือลดค่าตัวแปร)
    statement;
```

2. ลูป for ที่มีหลาย statement มีรูปแบบ ดังนี้

```
for (ตัวแปรและค่าเริ่มต้น; เงื่อนไขเปรียบเทียบ; เพิ่มหรือลดค่าตัวแปร)
{
    statement;
    statement;
    statement;
}
```

3. **ลูป for ซ้อนกัน** หมายถึง มีการใช้ statement ของ for อีกในขณะที่กำลังทำซ้ำด้วย for ไว้แล้ว โปรแกรมจะดำเนินการทำซ้ำในลูป โดยแต่ละรอบของลูปนอก จะต้องทำลูปในให้ครบทุกรอบก่อน แล้วจึงเริ่มทำลูปนอกในรอบต่อไป มีรูปแบบทั่วไป ดังนี้



- ตัวอย่างโปรแกรม *for\_exp1.cpp* แสดงการใช้ *for* แบบ *statement* เดียว แสดงข้อความซ้ำหลายครั้งบนจอภาพ โดยกำหนดค่าของตัวแปรตรวจสอบแบบเพิ่มค่าและลดค่าครั้งละ 1 หรือมากกว่า

```

/*Program : for_exp1.cpp
   Process : display for loop statement
*/
#include <iostream.h>
#include <conio.h>

void main()
{
    int i;
    clrscr();
    for(i=1;i<=5;i++) //set initial i value and condition of for loop
        cout<< "Program C++\n";
    getch();cout<<"\n";
    // condition with decrease value i = i-1
    for(i=5;i>=1;i--)
        cout<< "I love C++ Language\n";
    getch();cout<<"\n";
    for(i=16;i<=20;i++)

```

```

    cout<< "C++ Language is very easy for me\n";
    getch();
}

```

- ตัวอย่างโปรแกรม *for\_exp2.cpp* แสดงการใช้ *for* แบบหลาย *statement* คำนวณผลรวมและค่าเฉลี่ยจำนวน 1-10

```

/*Program : for_exp2.cpp
   Process : Calculate summation and everage 1-10, display all
*/
#include <iostream.h>
#include <conio.h>

void main()
{ float i,summation;
  float average;
  //begin statement
  clrscr();
  summation=0;average=0; // set initial value
  cout<< "Progarm display summation and average 1-10\n";
  for(i=1;i<=10;i++)
  { cout<<i<<'\n';
    summation=summation+i; //calculate summation 1-10
  }
  average=summation/10;
  cout<< "\nSummation 1-10 = "<<summation;
  cout<< "\nAverage 1-10  = "<<average;
  getch();
}

```

- ตัวอย่างโปรแกรม *for\_exp3.cpp* แสดงการหาผลรวมและค่าเฉลี่ย จำนวนตั้งแต่ 1-n โดยที่ผู้ใช้สามารถกรอกค่า *n* ในการคำนวณได้

```

/*Program : for_exp3.cpp
   Process : Calculate summation and everage 1-n, display all
*/
#include <iostream.h>
#include <conio.h>

void main()
{ float i,summation,n;
  float average;
  //begin statement
  clrscr();
}

```

```

summation=0;average=0; // set initial value
cout<< "Program display summation and average 1-n\n";
cout<< "Please enter n : ";cin>>n;
for(i=1;i<=n;i++)
{ cout<<i<<" ";
  summation=summation+i; //calculate summation 1-10
}
average=summation/n;
cout<< "\nSummation 1-10 = "<<summation;
cout<< "\nAverage 1-10  = "<<average;
getch();
}

```

- ตัวอย่างโปรแกรม *for\_exp4.cpp* แสดงการใช้ *for* แบบซ้อนกัน เพื่อคำนวณการคูณและแสดงผลลัพธ์

```

/*Program : for_exp4.cpp
Process : Calculate multiply number and display all
*/
#include <iostream.h>
#include <conio.h>

void main()
{ int i,j;

//begin statement
clrscr();
cout<< "Display Multiply calculation :\n";
for(i=2;i<=4;i++) //begin external loop
{ clrscr();
  cout<< "Display Multiply calculation of "<<i<<"\n";
  for (j=1;j<=12;j++) //begin internal loop
  {
    cout<<i<<"*"<<j<<" = "<<i*j;
    cout<<"\n";
  } //end of internal loop
  cout<< "press any key to continue...";
  getch();
} //end of external loop
cout<< "\nEnd of program...\a\n";getch();
}

```

## ◆ loop while

ลูป while จะเริ่มการทำงานด้วยการทดสอบเงื่อนไขที่กำหนดไว้ต้นลูปก่อนเสมอ ถ้าเงื่อนไขเป็นจริง จะทำงานซ้ำในลูป แต่ถ้าเงื่อนไขเป็นเท็จ จะเลิกทำงานในลูป (ทำงานในลูปขณะที่เงื่อนไขเป็นจริงเท่านั้น) รูปแบบของ while มีดังนี้

1. ลูป while ที่มี statement เดียว มีรูปแบบดังนี้

```
while(เงื่อนไขเปรียบเทียบ)
    statement;
```

2. ลูป while ที่มีหลาย statement มีรูปแบบดังนี้

```
while(เงื่อนไขเปรียบเทียบ)
{
    statement;
    statement;
    statement;
}
```

3. ลูป while ซ้อนกัน มีรูปแบบ ดังนี้

```
while(เงื่อนไขเปรียบเทียบของลูปนอก)
{
    statement;
    statement;
    while(เงื่อนไขเปรียบเทียบของลูปใน)
    {
        statement;
        statement;
        statement;
    }
    statement;
    statement;
}
```

- ตัวอย่างโปรแกรม `while1.cpp` แสดงการกำหนดเงื่อนไข เพื่อให้สามารถพิมพ์ข้อความซ้ำ ๆ ได้ จำนวน 5 ครั้ง ดังนี้

```

/*Program : while1.cpp
   Process : display text 5 times
*/
#include <iostream.h>
#include <conio.h>
void main()
{ int i;
  clrscr();
  i=1; //set initial of variable in while conditio
  while(i<=5)
  { cout<< "I love C++ Langauge\n";
    i++; // increase variable value
  }
  getch();
}

```

- ตัวอย่างโปรแกรม `while2.cpp` แสดงการทำงานของโปรแกรมโดยใช้ `while` แบบ หลาย `statement` โดยโปรแกรมจะให้พิมพ์ตัวอักษรที่ถูกต้อง ซึ่งมีอยู่ 3 ตัว คือ 8, \* และ a ตัวใดตัวหนึ่ง ถ้าพิมพ์อักษรไม่ตรงตามที่กำหนดไว้ โปรแกรมจะให้พิมพ์ใหม่ทางคีย์บอร์ด ดังนี้

```

/*Program : while2.cpp
   Process : enter correct character
*/
#include <iostream.h>
#include <conio.h>

void main()
{ char ch;
  int count;

  //begin statement
  clrscr();
  count=0;
  cout<< "Please enter character : \n";
  while (ch!='8' && ch!='*' && ch!='a' )
  {cin>>ch; //enter character from keyboard
    count++; //count of times that enter character
  }
}

```

```

cout<< "Character "<<ch<< " is correct !!! \n\n";
cout<<"You enter "<<count<<" times";
getch();
}

```

- ตัวอย่างโปรแกรม *while3.cpp* แสดงการคำนวณผลรวมและหาค่าเฉลี่ยของจำนวน 1 - 10 ด้วยการใช้อุปกรณ์ *while* ที่ให้ผลลัพธ์เหมือนการใช้อุปกรณ์ *for*

```

/*Program : while4.cpp
Process : calculate summation and average 1-10*/
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
void main()
{ float number,sum,average;
//begin statment
clrscr();
cout<< "Program calculate summation and average 1-10\n";
sum=0; average=0; number=1;
while (number<=10) //begin loop
{ cout<<number<<"\n";
sum+=number; //same as ... sum=sum+number
number++; //increase variable +1
} //end loop
average=sum/(number-1);
cout<< "\nSummation number 1-10 = "<<<sum;
cout<< "\nAverage number 1-10= "<<<setprecision(2)<<average;
getch();
}

```

## ◆ ลูป do

ลูป do จะเริ่มด้วยการทำงาน *รอบแรก 1 รอบก่อนเสมอ* และมีการทดสอบเงื่อนไขที่ท้ายลูป ถ้าเงื่อนไขเป็น *จริง* จะมีการทำซ้ำในลูป แต่ถ้าเงื่อนไขเป็น *เท็จ* จะเลิกทำงานในลูป มีรูปแบบดังนี้

1. ลูป do ที่มี statement เดียว มีรูปแบบดังนี้

```
do
    statement;
while(เงื่อนไขเปรียบเทียบ);
```

2. ลูป do ที่มีหลาย statement มีรูปแบบดังนี้

```
do{
    statement;
    statement;
    statement;
}while(เงื่อนไขเปรียบเทียบ);
```

3. ลูป do ซ้อนกัน มีรูปแบบดังนี้

```
do{
    statement;
    statement;
    do{
        statement;
        statement;
    }while(เงื่อนไขเปรียบเทียบลูปใน);
    statement;
}while(เงื่อนไขเปรียบเทียบลูปนอก);
```



- ตัวอย่างโปรแกรม *dowhile1.cpp* แสดงการพิมพ์ข้อความ *I love C++ Langauge* ซ้ำจำนวน 5 ครั้ง โดยใช้รูปแบบหลาย *statement*

```

/*Program : dowhile1.cpp
   Process : display text 5 times*/
#include <iostream.h>
#include <conio.h>
void main()
{ int i;
  clrscr();
  i=1; //set initial of variable in do...while condition
  //begin loop
  do{
    cout<< "I love C++ Langauge\n";
    i++; //increase value of i variable +1
  }while(i<=5); //condition for loop
  getch();
}

```

- โปรแกรม *dowhile2.cpp* แสดงการคำนวณผลรวมและค่าเฉลี่ย 1-10 โดยใช้รูปแบบ *do...while* ซึ่งให้ผลลัพธ์เหมือนกับการใช้รูปแบบ *while*

```

/*Program : dowhile2.cpp
   Process : calculate summation and average 1-10
*/
#include <iostream.h>
#include <conio.h>
void main()
{ float number,sum,average;
  //begin statment
  clrscr();
  cout<< "Program calculate summation and average 1-10\n";
  sum=0; average=0;
  number=1;
  do{
    cout<<number<<'\n';
    sum+=number; //calculate summation
    number++;

```

```

    }while (number<=10);
average=sum/(number-1);
cout<< "\nSummation number 1-10 = "<<sum;
cout<< "\nAverage number 1-10= "<<average;
getch();
}

```

### ◆ การออกจากลูปด้วย break

**break** เป็นคีย์เวิร์ด(keyword) เพื่อให้เลิกการทำงานหรือออกจากลูปทันที ในขณะที่การทดสอบเงื่อนไขที่กำหนดไว้ ยังไม่ทำให้ค่าเปรียบเทียบที่มีผลให้ต้องออกจากลูป สามารถนำ break ไปใช้ได้กับลูปทุกประเภท (โดยส่วนใหญ่ break ใช้ร่วมกับประโยคที่มีเงื่อนไขมาก่อน เช่น if ตัวอย่างการใช้ break กับลูป for เป็นดังนี้

```

for (ตัวแปรและค่าเริ่มต้น; เงื่อนไขเปรียบเทียบ; เพิ่มหรือลดค่าตัวแปร)
{
    statement;
    statement;
    break;          ← ออกจากลูป
    statement;
}
←

```

- ตัวอย่างโปรแกรม *break.cpp* แสดงการทำงานหาผลรวมของลูป *for* ของเลข 1-100 แต่มีเงื่อนไขว่า ถ้าค่าของตัวแปร *row* มากกว่า 23 ให้ *break* ออกจากลูปก่อนที่จะครบ

100 รอบ

```

/*Program : break.cpp
Process : display using 'break' in for loop*/
#include <iostream.h>
#include <conio.h>
void main()
{ int x,row,sum;
  clrscr();
  row=0;sum=0;
  for(x=1;x<=100;++x) //condition for 100 loop
  { cout<<"x="<<x<<"\n";
    sum+=x;
    row++;

```

```

if(row>23) //condition of break
    break;
}
cout<<"summation 1-"<<x<<" = "<<sum; getch();
}

```

### ◆ การย้อนกลับไปต้นลูปด้วย continue

**continue** เป็นคีย์เวิร์ด (keyword) เพื่อกำหนดให้โปรแกรมย้อนกลับไปเริ่มทำงานที่ต้นลูปนั้นใหม่ ในขณะที่ยังทำงานไม่ถึงท้ายลูป หรือยังไม่ครบจำนวนรอบที่กำหนดไว้ในเงื่อนไข สามารถใช้ continue กับลูป for while do...while ได้ทุกประเภท

- ตัวอย่างโปรแกรม *continue.cpp* แสดงการใช้ *continue* ให้กลับไปเริ่มทำงานที่ต้นลูปใหม่

```

/*Program : continue.cpp
Process : display keyword 'continue' in do...while loop */
#include<iostream.h>
#include<conio.h>

void main()
{ float x,y;
  char choice;
  do{
    clrscr();
    cout<<"\nProgram Divide Calculation x/y";
    cout<<"\n*****";
    cout<<"\nEnter X : "; cin>>x;
    cout<<"\nEnter Y : "; cin>>y;
    if(y==0)
    { cout<<"\a\n\nCan't divide by zero !!!";
      cout<<"\npress any key to continue...";
      getch();
      continue; //return to begin loop
    }
    cout<<"\n*** Result "<<x<<"/"<<y<<" = "<<x/y;
    cout<<"\n\nCalculate another <y or n> ? ";
    cin>>choice;
  }while((choice!='n')&&(choice!='N'));
  clrscr();
}

```

### ◆ แบบฝึกหัดท้ายบท

1. ให้เขียนโปรแกรมเพื่อคำนวณหาค่า  $n!$  (  $n$  factorial) โดยกรอกค่า  $n$  ทางคีย์บอร์ดได้
2. ให้เขียนโปรแกรมเพื่อแสดงตัวอักษร(character) จากระหัส ASCII ตั้งแต่ 0 - 255
3. ให้เขียนโปรแกรมแสดงจำนวนเลขคี่(odd) ตั้งแต่ 1-1000 คำนวณหาผลรวมและค่าเฉลี่ยของจำนวนคี่ พร้อมกับนับว่ามีจำนวนคี่กี่จำนวนที่หารด้วย 3 และ 4 ลงตัว มีจำนวนใดบ้าง และมีกี่จำนวน
4. ให้เขียนโปรแกรมหาจำนวนที่ 3 และ 6 หารลงตัว ระหว่าง 500 – 1000 พร้อมกับนับว่ามีกี่จำนวน รวมถึงคำนวณหาผลรวมและค่าเฉลี่ยของจำนวนดังกล่าว
5. ให้เขียนโปรแกรมเพื่อแสดงตารางการคูณ โดยสามารถกำหนดจำนวนเริ่มต้น และจำนวนสุดท้ายได้ทางคีย์บอร์ด เช่น

Begin : 2

End : 4

#### Result

|         |         |         |
|---------|---------|---------|
| 2*1=2   | 3*1=3   | 4*1=2   |
| 2*2=4   | 3*2=6   | 4*2=8   |
| 2*3=6   | 3*3=9   | 4*3=12  |
| .....   | .....   | .....   |
| 2*12=24 | 3*12=36 | 4*12=48 |

6. ให้เขียนโปรแกรมเพื่อแสดงจำนวนตัวเลข และหาผลรวมของจำนวนคู่และคี่ ให้มีรูปแบบการแสดงผลดังนี้

|   |   |   |   |    |    |    |    |    |    |
|---|---|---|---|----|----|----|----|----|----|
| 1 | 3 | 5 | 7 | 9  | 11 | 13 | 15 | 17 | 19 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |

SUMMATION OF ODD NUMBER = xxx

SUMMATION OF EVEN NUMBER = xxx

7. จงเขียนโปรแกรมแสดงการคำนวณเลขยกกำลังสอง จำนวน 1 -10 ให้มีรูปแบบการแสดงผลดังนี้

|     |      |
|-----|------|
| 1   | 1    |
| 2   | 4    |
| 3   | 9    |
| 4   | 16   |
| ... | .... |
| 10  | 100  |

8. ให้เขียนโปรแกรมคำนวณการคูณ ของ 1 -9 ให้มีรูปแบบการแสดงผล ดังนี้

|   | 1 | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|---|---|----|----|----|----|----|----|----|----|
| 1 | 1 |    |    |    |    |    |    |    |    |
| 2 | 2 | 4  |    |    |    |    |    |    |    |
| 3 | 3 | 6  | 9  |    |    |    |    |    |    |
| 4 | 4 | 8  | 12 | 16 |    |    |    |    |    |
| 5 | 5 | 10 | 15 | 20 | 25 |    |    |    |    |
| 6 | 6 | 12 | 18 | 24 | 30 | 36 |    |    |    |
| 7 | 7 | 14 | 21 | 28 | 35 | 42 | 49 |    |    |
| 8 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 |    |
| 9 | 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 |

9. ให้เขียนโปรแกรมแสดงค่าจำนวนเฉพาะ(prime) มีช่วงข้อมูลระหว่าง 1- 300 พร้อมกับนับว่ามีจำนวนเฉพาะ(prime) กี่จำนวน

10. สมมติให้ I, J, K เป็นตัวแปรประเภท integer ให้เขียนรูปแบบผลลัพธ์ของ โปรแกรมที่กำหนดให้ ดังนี้

```

10.1 I=0;
    J=0;
    for(;;)
    {
        K=2*I*J;
        if(K>10)
            break;
        cout<<I<<J<<K<<'\n';
        I++;
        J++;
    }
    cout<<K<<'\n';

```

```

10.2 I=0;
    J=0;
    for(;;)
    {   K=2*I*J;
        if(K>10)
            break;
        cout<<I<<J<<K<<'\n';
        if(I+J>5)
            break;
        I++;
        J++;
    }
    cout<<K<<'\n';

```

```

10.3 I=5;
    for(;;)
    {   cout<<I;
        I -=2;
        if(I<1)
            break;
        J=0;
        for(;;)
        {   J++;
            cout<<J;
            if(J>=I)
                break;
        }
        cout<<"###\n";
    }
    cout<<"***\n";

```

11. Write a program to read a sequence of positive integers and to print the greatest of them. Use a negative to signal the end of the input data.
12. Write a program that reads a sequence of positive real numbers and computes their average. A negative number to signal the end of the input data.