

บทที่ 4

คำสั่งควบคุม(Control Statements)

วัตถุประสงค์

1. เพื่อให้ นักศึกษา รู้จัก รูปแบบ คำสั่งการควบคุม (Control Statement) แบบ Sequential, Selection และ Loop
2. เพื่อให้ นักศึกษา สามารถ สร้าง และ ใช้ งาน คำสั่งการควบคุม ทิศทาง การทำงาน ของ โปรแกรม ได้

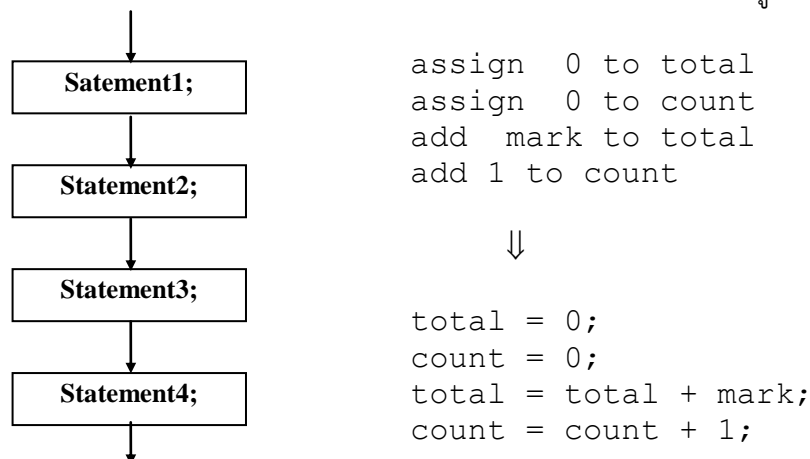
คำสั่งควบคุมการทำงานของโปรแกรม(Control Flow หรือ Control Statement)

ลักษณะการควบคุมการทำงานของโปรแกรมสามารถแบ่งออกได้เป็น 3 แบบ คือ

1. แบบเรียงลำดับ (Sequential) เป็นลักษณะคำสั่งที่จัดวางเรียงลำดับไว้ โดยทำงานตั้งแต่คำสั่งแรกไปจนจบโปรแกรม
2. แบบทางเลือก (Selection) เป็นการทำงานของคำสั่งแบบเลือกทำ โดยมีเงื่อนไขในการตัดสินใจว่าจะทำคำสั่งใดต่อไปหรือจะไปในทิศทางใด
3. แบบวนรอบ (Loop) เป็นการทำงานของคำสั่งแบบทำซ้ำ โดยสามารถระบุจำนวนครั้งของคำสั่งที่ทำงานได้แน่นอน หรือไม่ทราบจำนวนแต่ใช้เงื่อนไขเป็นการตัดสินใจ

คำสั่งการควบคุมแบบเรียงลำดับ (Sequential)

เป็นลักษณะการทำงานของคำสั่งที่จัดวางเรียงลำดับการทำงานไว้เท่านั้น ไม่มีรูปแบบคำสั่ง



ตัวอย่าง

```

int num;
cout << "Enter number : ";
cin >> num;
num = num + 10;
cout << "Square of " << num << " = " << num*num;
return(0);

```

คำสั่งการควบคุมแบบเลือกทำ (Selection)

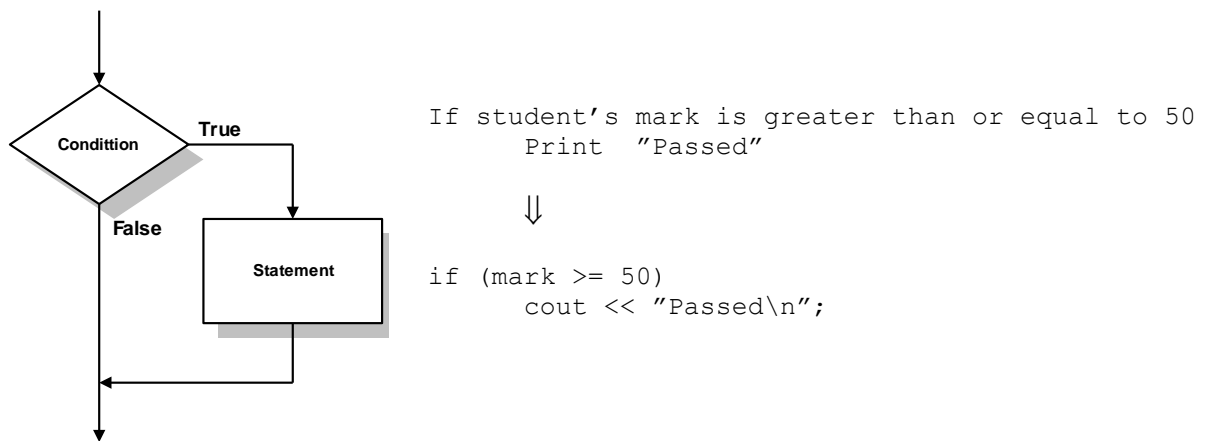
เป็นรูปแบบให้เลือกการทำงานของคำสั่งตามเงื่อนไขที่กำหนดไว้ โดยการสร้างเงื่อนไขด้วยตัวดำเนินการเปรียบเทียบและตรรกศาสตร์ มีคำสั่งที่ใช้งานดังนี้ คือ

■ if Statement

เป็นคำสั่งที่ทำงานโดยการตรวจสอบเงื่อนไขที่กำหนดว่ามีค่าเป็นจริงหรือเท็จ แล้วจึงทำงานตามคำสั่งที่ได้กำหนดไว้ สามารถใช้กับข้อมูลได้ทั้งตัวเลข ตัวอักษร และสตริง สำหรับคำสั่งที่ให้ทำงานเป็นได้ทั้ง simple statement หรือ compound statement สามารถแบ่งการใช้งานได้เป็น 3 แบบ คือ

รูปแบบที่ 1 ใช้ if อย่างเดียว เมื่อต้องการทำงานในกรณีที่เงื่อนไขเป็นจริงเท่านั้น ส่วนเท็จไม่มีการทำงาน

if (conditions) statement;



ตัวอย่าง

```

if (x < 0) cout << x;

if (pastdue > 0) credit = 0;

if (x <= 3.0) {
    y = 3 * sqrt(x);
    cout << y << endl;
}

if ((balance < 1000.0) || (status == 'R'))
    cout << balance;

if ((a >= 0) && (b <= 5)) {
    xmid = (a+b) /2;
    ymid = sqrt(xmid);
}
  
```

การทดลองที่ 4_1 โปรแกรมแสดงการทำงานของคำสั่ง if

```

1  /* Program 4_1 : Use if statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int A, B;
7      cout << "Enter value A and B : ";
8      cin >> A >> B;
9      if (A == B) cout << "A and B values are equal" << endl;
10     if (A > B) cout << "A values > values B" << endl;
11     if (A < B) cout << "A values < values B" << endl;
12     return (0);
13 }

```

บันทึกผลการทดลอง

รูปแบบที่ 2 ใช้ if – else เมื่อมีทำงานทั้งกรณีที่เงื่อนไขเป็นจริงหรือเท็จ โดยหากเงื่อนไขมีค่าเป็นจริงให้ทำคำสั่งหลัง if แต่ถ้ามีค่าเป็นเท็จให้ทำคำสั่งหลัง else

```

if (conditions) statement1;
else statement2;

```

```

if student's mark is greater than or equal to 50
    Print "Passed"
else
    Print "Failed"

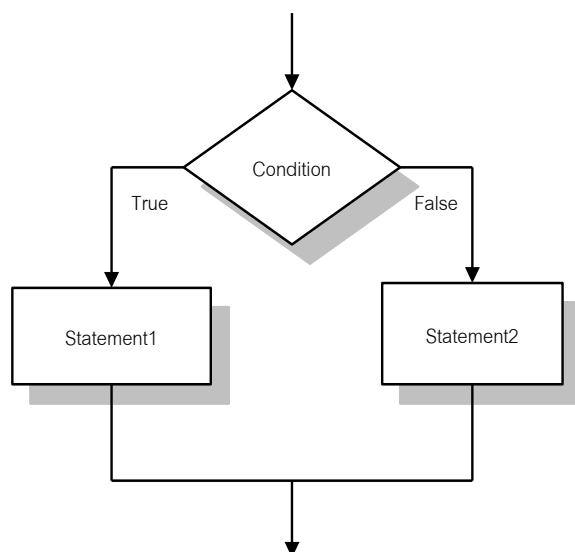
```

⇓

```

if (mark >= 50)
    cout << "Passed";
else
    cout << "Failed";

```



ตัวอย่าง

```

if (status == 'S') tax = 0.20*pay;
else tax = 0.14*pay;

if (pastdue > 0) {
    cout << "Account number " << account_no << " is overdue";
    credit = 0;
}
else credit = 1000.0;

if (circle) {
    cin >> radius;
    area = 3.14159*radius*radius;
    cout << "Area of circle = " << area;
}
else {
    cin >> length >> width;
    area = length * width;
    cout << "Area of rectabgle = " << area;
}

```

การทดลองที่ 4_2 โปรแกรมแสดงการทำงานของคำสั่ง if -else

```

1  /* Program 4_2 : Use if-else statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int year,maxday;
7      cout << "\nEnter year : ";
8      cin >> year;
9      if ( (year % 4) == 0) { /* Condition True */
10         cout << "In " << year << " is leap year ";
11         maxday = 29;
12     }
13     else { /* Condition False */
14         cout << "In " << year << " is not leap year ";
15         maxday = 28;
16     }
17     cout << "and February has " << maxday << " day.\n";
18     return(0);
19 }

```

บันทึกผลการทดลอง

คำถาม จากโปรแกรมที่ 4_2 ให้นักศึกษาเปลี่ยนการใช้คำสั่ง if-else เป็น ตัวดำเนินการเงื่อนไข (? :) ทำงานแทน โดยผลลัพธ์ของโปรแกรมยังคงเหมือนเดิม (ให้แก้ไขเฉพาะคำสั่ง if-else เท่านั้น สามารถใช้ตัวดำเนินการเงื่อนไขมากกว่าหนึ่งครั้งได้)

รูปแบบที่ 3 ใช้แบบ nested-if เป็นการทำงานแบบ if ซ้อน if โดยจะมีจำนวนกี่ครั้งก็ได้

```
if (condition1) statement1;
else if (condition 2) statement2;
else statement3;
```

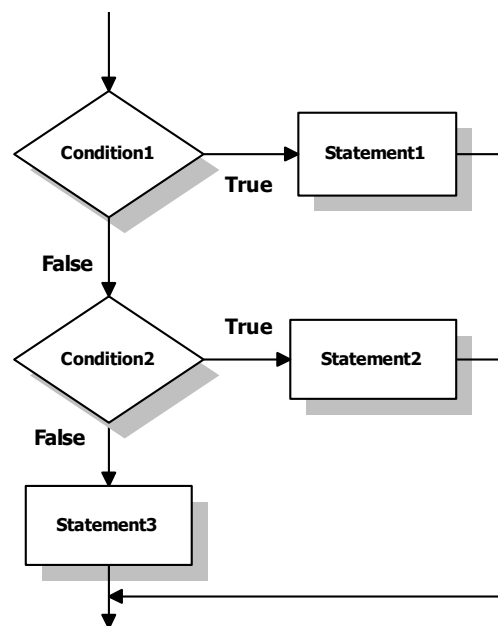
รูปแบบการใช้งาน nested-if ในลักษณะต่างๆ

```
if e1 if e2 s1;
      else s2;
else if e3 s3;
      else s4;
```

```
if e1 s1;
else if e2 s2;
      else s3;
```

```
if e1 if e2 s1;
      else s2;
else s3;
```

```
if e1 {
    if e2 s1;
}
else s2;
```



ตัวอย่าง

```

if (mark >= 80) cout << "A";
else if (mark >= 70) cout << "B";
else if (mark >= 60) cout << "C";
else if (mark >= 50) cout << "D";
else printf("F");

if ((time >= 0.00) && (time < 24.00)) {
    if ((time >= 0.00) && (time < 12.00))
        cout << "Good Morning";
    else if ((time >= 12.00) && (time < 18.00))
        cout << "Good Afternoon";
    else if ((time >= 18.00) && (time < 24.00))
        cout << "Good Evening";
}
else cout << "Time is out of range";

```

การทดลองที่ 4_3 โปรแกรมแสดงการทำงานของคำสั่ง nested if -else

```

1  /* Program 4_3 : Use nested if-else statement */
2  #include <iostream>
3  #include <string>
4  using namespace std;
5  int main()
6  {
7      char ch;
8      cout << "\nEnter a character : ";
9      cin >> ch;
10     cout << endl;
11     if (ch >= 'a' && ch <= 'z')
12         cout << ch << " is lower character.";
13     else if (ch >= 'A' && ch <= 'Z')
14         cout << ch << " is upper character.";
15     else if (ch >= '0' && ch <= '9')
16         cout << ch << " is numeric.";
17     else cout << ch << " is special character.";
18     cout << endl;
19     return(0);
20 }

```

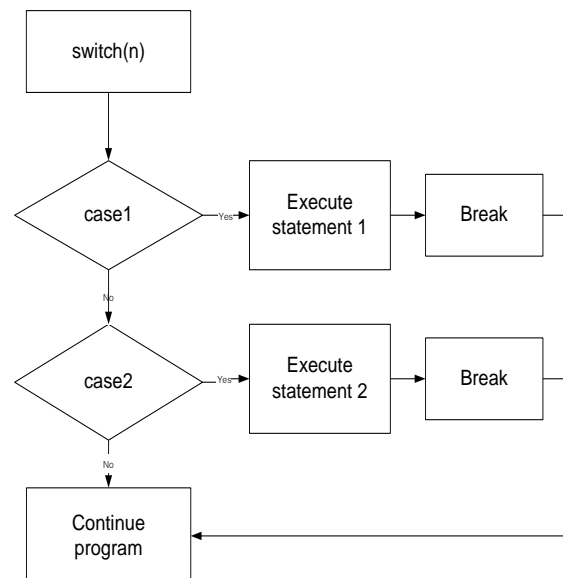
บันทึกผลการทดลอง

■ Switch Case

คำสั่ง Switch ใช้เพื่อเลือกการทำงานของกลุ่มคำสั่งใดกลุ่มหนึ่งจากหลายๆกลุ่มที่เป็นไปได้ หรือที่กำหนดไว้ สามารถใช้ได้กับข้อมูลที่เป็นตัวเลขจำนวนเต็ม และตัวอักษร 1 ตัว ซึ่งการเลือกกลุ่มจะขึ้นอยู่กับเงื่อนไขที่กำหนดไว้เป็นค่าคงที่

รูปแบบ

```
switch(expression)
{
    case constant1:
        statement sequence;
        break;
    case constant2:
        statement sequence;
        break;
    :
    default:
        statement sequence;
}
```



```
switch( choice )
{
    case 'R' :    cout << "RED";
                  break;
    case 'W' :    cout << "WHITE";
                  break;
    case 'B' :    cout << "BLUE";
                  break;
    case 'G' :    cout << "GREEN";
                  break;
    default:      cout << "I don' t know.";
}

```

การทดลองที่ 4_4 โปรแกรมแสดงการทำงานของคำสั่ง switch case

```

1  /* Program 4_4 : Use nested switch case statement */
2  #include <iostream>
3  #include <string>
4  using namespace std;
5  int main()
6  {
7      cout << "Please enter a simple expression ";

```

```

8      cout << "(number operator number) : ";
9      int LeftOperand, RightOperand;
10     char Operator;
11     cin >> LeftOperand >> Operator >> RightOperand;
12     int Result;
13     switch (Operator) {
14         case '+': Result = LeftOperand + RightOperand;
15                 break;
16         case '-': Result = LeftOperand - RightOperand;
17                 break;
18         case '*': Result = LeftOperand * RightOperand;
19                 break;
20         case '/': Result = LeftOperand / RightOperand;
21                 break;
22         default: cout << Operator << " is unrecognized operation.";
23                 cout << endl;
24                 return(1);
25     }
26     // Display result
27     cout << LeftOperand << " " << Operator << " ";
28     cout << RightOperand << " equals " << Result << endl;
29     return(0);
30 }

```

บันทึกผลการทดลอง

คำถาม จากโปรแกรมที่ 4_4 ให้นักศึกษาเปลี่ยนการใช้คำสั่ง switch เป็น if ทำงานแทน โดยผลลัพธ์ของโปรแกรมยังคงเหมือนเดิม (ให้แก้ไขเฉพาะคำสั่ง switch เท่านั้น)

การทดลองที่ 4_5 โปรแกรมแสดงการทำงานของการทำงานของการคำนวณหาพื้นที่

```
1  /* Program 4_5 : Calculate Area */
2  #include <iostream>
3  #include <string>
4  using namespace std;
5  int main()
6  {
7      char Choice;
8      float Area;
9      cout << " Program Calculate Area " << endl;
10     cout << " 1. Circle" << endl;
11     cout << " 2. Rectangle" << endl;
12     cout << " 3. Exit" << endl;
13     cout << "Enter your choose number: ";
14     cin >> Choice;
15     if (choice == '1') {
16         float Radius;
17         cout << "\nEnter radius : ";
18         cin >> Radius;
19         Area = 3.14159F * Radius * Radius;
20         cout << "Area of Circle = " << Area << endl;
21     }
22     else if (Choice == '2') {
23         float Length,Width;
24         cout << "Enter length and width : ";
25         cin >> Length >> Width;
26         Area = Length * Width;
27         cout << "Area of Rectangle = " << Area << endl;
28     }
29     else if (Choice == '3') cout << "\n . . . Exit Program . . .\n";
30     else cout << "\nYou choose out of range is not process.\n";
31     return(0);
32 }
```

บันทึกผลการทดลอง

คำสั่งการควบคุมแบบแบบวนรอบ

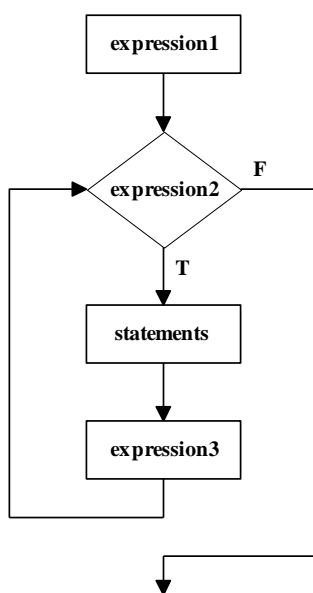
สำหรับคำสั่งการวนรอบเป็นรูปแบบการทำงานคำสั่งที่ซ้ำหลายๆ ครั้ง ทั้งที่มีจำนวนครั้งแน่นอน และไม่แน่นอน ซึ่งจะทำงานตามเงื่อนไขที่กำหนดไว้ โดยสร้างเงื่อนไขด้วยตัวดำเนินการเปรียบเทียบและตรรกศาสตร์ มีคำสั่งที่ใช้งานดังนี้ คือ

■ for statement

คำสั่ง for เป็นคำสั่งควบคุมการวนรอบ ประกอบด้วยนิพจน์สำหรับกำหนดค่าเริ่มต้นให้กับดัชนี (index), นิพจน์ที่ใช้ตรวจสอบว่าจะทำการวนรอบต่อไปหรือไม่ และนิพจน์ที่ใช้เปลี่ยนค่าดัชนีในแต่ละรอบการทำงาน

รูปแบบ

for (expression1 ; expression2 ; expression3) statement;



expression1 เป็นการกำหนดค่าเริ่มต้นให้กับพารามิเตอร์

expression2 เป็นการทดสอบเงื่อนไขของการวนรอบ

expression3 เป็นนิพจน์แบบยูนิควารี หรือการกำหนดค่า

```
int digit;
for (digit = 0; digit <= 9; ++digit)
    cout << digit << endl;
```

```
for (int n = 0; n <= 20; n++)
    cout << n << endl ;
```

```
for (int n = 10; n >= 1; n--)
    cout << n << endl ;
```

```
int sum = 0;
for (int n = 10; n >= 1; n--) {
    cout << n << endl;
    sum = sum + n;
}
```

การทดลองที่ 4_6 โปรแกรมแสดงการทำงานของคำสั่ง for

```
1  /* Program 4_6 : Use for statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      // Use sequential structure
7      cout << 1 ;
8      cout << 2 ;
9      cout << 3 ;
10     cout << 4 ;
11     cout << 5 ;
12     cout << endl;
13     // Use loop structure
14     for (int n = 1 ; n <= 5 ; n++)      cout << n;
15     cout << endl;
16     return(0);
17 }
```

บันทึกผลการทดลอง

การทดลองที่ 4_7 โปรแกรมแสดงการทำงานของคำสั่ง for

```
1  /* Program 4_7 : Use for statement */
2  #include <iostream>
3  #include <string>
4  using namespace std;
5  int main()
6  {
7      int Number;
8      string Name;
9      cout << "Enter String : ";
10     cin >> Name;
11     cout << "Enter number : ";
12     cin >> Number;
13     for (int n = 1 ; n <= Number ; n++)
14         cout << n << "." << Name << endl;
15     cout << endl;
16     return(0);
17 }
```

บันทึกผลการทดลอง

คำถาม จากโปรแกรมที่ 4_7 ให้นักศึกษาเปลี่ยนการใช้คำสั่ง for จากการเพิ่มค่า เป็นการลดค่าลงแทน โดยผลลัพธ์ของโปรแกรมยังคงเหมือนเดิม (ให้แก้ไขเฉพาะคำสั่ง for เท่านั้น)

การทดลองที่ 4_8 โปรแกรมแสดงการทำงานของคำสั่ง for

```

1  /* Program 4_8 : Use for statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int Num;
7      cout << "Enter number : ";
8      cin >> Num;
9      int Sum = 0;
10     int Value;
11     for (int n = 1 ; n <= Num ; n++) {
12         cout << "Enter integer value(" << n << ") : ";
13         cin >> Value;
14         Sum = Sum + Value;
15     }
16     cout << "\nSummation of integer " << Num;
17     cout << " value = " << Sum << endl;
18     return(0);
19 }
```

บันทึกผลการทดลอง

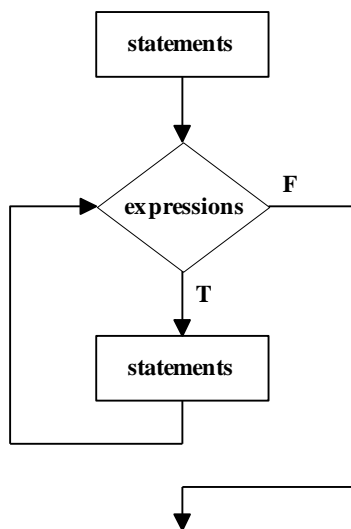
■ while statement

คำสั่ง while ใช้เพื่อควบคุมการวนรอบ โดยมีตรวจสอบเงื่อนไขก่อนแล้วค่อยทำคำสั่งที่กำหนด มีรูปแบบทั่วไปดังนี้

รูปแบบ while (expression) statement;

expression เป็นเงื่อนไขที่สร้างไว้เพื่อทำงานโดยหากมีค่าเป็นจริงจะวนรอบและทำคำสั่งที่กำหนดใน statement เมื่อเป็นเท็จจะเลิกออกจากลูป

statement เป็นคำสั่งที่กำหนดให้ทำงานเป็นได้ทั้งคำสั่ง simple และ compound



```

int digit = 0;
while (digit <= 9)
{
    cout << digit;
    digit++;
}
  
```

```

int digit = 20;
while (digit > 10)
{
    cout << digit;
    digit--;
}
  
```

การทดลองที่ 4_9 โปรแกรมแสดงการทำงานของคำสั่ง while แบบจำนวนรอบคงที่

```

1  /* Program 4_9 : Use while statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      cout << "Display square of number 1 to 10.\n";
7      cout << "=====\n";
8      cout << "\tX\tX^2\n";
9      cout << "=====\n";
10     int    X = 1;
11     // display value power 2
12     while (X <= 10) {
13         cout << "\t" << X << "\t" << X*X << endl;
14         X++;
15     }
16     return(0);
17 }
  
```

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

```

1  /* Program 4_10 : Use while statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int ListSize = 5;
7      int ValuesProcessed = 0;
8      float ValueSum = 0;
9      cout << "Please enter " << ListSize << " numbers " << endl;
10     while (ValuesProcessed < ListSize) {
11         float Value;
12         cin >> Value;
13         ValueSum += Value;
14         ++ValuesProcessed;
15     }
16     float Average = ValueSum / ValuesProcessed;
17     cout << "Average : " << Average << endl;
18     return(0);
19 }
20

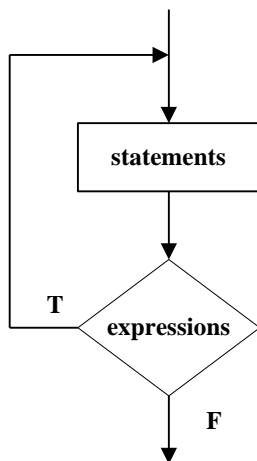
```

[illegible]

เป็นคำสั่งในการวนรอบ โดยจะทำคำสั่งก่อนหนึ่งครั้ง แล้วจึงค่อยตรวจสอบเงื่อนไข มีรูปแบบทั่วไปดังนี้

รูปแบบ do statement while (expression);

statement เป็นคำสั่งที่กำหนดเพื่อให้ทำงาน เขียนได้เป็นทั้งคำสั่ง simple และ compound



```
int    digit = 10;

do {
    cout << digit;
    digit--;
} while (digit >= 1);
```

การทดลองที่ 4_11 โปรแกรมแสดงการทำงานของคำสั่ง do while

```

1  /* Program 4_11 : Use do while statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      unsigned long Sum = 0L;
7      int n, i = 1;
8      cout << "Please enter value n to find sumation of n*n : ";
9      cin >> n;
10     do {
11         Sum += i * i;
12         i++;
13     } while (i <= n);
14     cout << "Sum = " << Sum << endl;
15     return(0);
16 }
17

```

บันทึกผลการทดลอง

คำถาม จากโปรแกรมที่ 4_11 ให้นักศึกษาอธิบายว่าทำไมคำสั่งบรรทัดที่ 7 การให้ค่าตัวแปร Sum ต้องมีตัวอักษร L ต่อท้าย

การทดลองที่ 4_12 โปรแกรมแสดงการทำงานของคำสั่ง do while

```

1  /* Program 4_12 : Use do while statement */
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int Num, Value, Sum;
7      bool Flag = true;
8      Num = 1;
9      Sum = 0;
10     do {
11         cout << "Enter value " << Num << " : ";
12         cin >> Value;
13         if (Value > 0) {
14             Sum += Value;
15             Num++;
16         }
17     }
18 }

```



```

17         else Flag = false;
18     }while ( Flag == true);
19     if (Num > 1) {
20         float Avg;
21         Avg = static_cast<float>(Sum)/--Num;
22         cout << "Average of data = " << Avg << endl;
23     }
24     else cout << "No data input." << endl;
25     return(0);
26 }
27

```

บันทึกผลการทดลอง

การทดลองที่ 4_13 โปรแกรมสำหรับคำนวณหาค่าแฟคทอเรียล

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int Number, Count;
6      bool Flag = true;
7      /* initialize and read value for number */
8      do {
9          cout << "\nInput number factorial(0-20)? : ";
10         cin >> Number;
11         if ((Number >= 0) &&(Number <= 20)) Flag = false;
12         else cout << "Number value over limit.\n";
13     } while ( Flag );
14     /* calculate factorial of number */
15     if (Number >= 0) {
16         long Fac = 1;
17         if (Number > 0) {
18             Count = 1;
19             while (Count <= Number) Fac *= Count++;
20             cout << "\nFactorial of " << Number << " = ";
21             cout << Fac << endl;
22         }
23     }
24     return(0);
25 }

```

บันทึกผลการทดลอง

การทดลองที่ 4_14 โปรแกรมทำการนับจำนวนตัวอักษรที่คีย์เข้ามา

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main()
5  {
6      char Ch;
7      int Count = 0;
8      string Str;
9      cout << "Enter String : ";
10     while (cin >> Ch) {
11         Count++;
12         Str += Ch;
13     }
14     cout << "\nLength of string(" << Str << ") is ";
15     cout << Count << endl;
16     return(0);
17 }
```

บันทึกผลการทดลอง

คำถาม จากโปรแกรมที่ 4_14 ให้นักศึกษาเปลี่ยนรูปแบบการรับค่าโดยใช้ฟังก์ชัน get ในการทำงานสำหรับการตรวจสอบเงื่อนไขของคำสั่ง while เป็นดังนี้ (Ch = cin.get()) != '\n' แทน จากนั้นให้ทดลองรันโปรแกรมสังเกตผลลัพธ์

การทดลองที่ 4_15 โปรแกรมรับข้อความและแสดงข้อความทีละอักขระ

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main()
5  {
6      string Str;
7      cout << "Enter String : ";
8      cin >> Str;
9      cout << endl;
10     cout << "Your string enter : " << Str << endl;
11     cout << "Length of string = " << Str.length() << endl;
12     for (int n = 0 ; n < Str.length() ; n++) {
13         cout << "Position " << n << " = " << Str.at(n) << endl;
14     }
15     return(0);
16 }

```

บันทึกผลการทดลอง

คำถาม จากโปรแกรมที่ 4_15 ในบรรทัดที่ 8 ให้นักศึกษาเปลี่ยนการทำงานเป็นฟังก์ชันชื่อ getline ทำงานแทนดังนี้ getline(cin, Str, '\n') จากนั้นให้ทดลองรันโปรแกรม สังเกตผลลัพธ์

แบบฝึกหัดท้ายบท

1. จงเขียนโปรแกรมสำหรับคำนวณเกรดที่ได้ โดยมีการรับข้อมูลเป็นค่าคะแนนจำนวนเต็มอยู่ในช่วง 0 ถึง 100 และให้มีการตรวจสอบความถูกต้องของคะแนนที่ต้องอยู่ระหว่าง 0 ถึง 100 เท่านั้น หากเป็นค่าอื่นให้แจ้งข้อความผิดพลาด ซึ่งมีการแบ่งช่วงคะแนนของเกรด ดังนี้

คะแนน	เกรด
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
0 - 59	F

2. จงเขียนโปรแกรมแสดงผลเป็นรูปสามเหลี่ยม โดยมีการรับตัวเลขจำนวนเต็มบวกที่มีค่าไม่เกิน 20 และรับตัวอักษรหนึ่งตัวที่ต้องการแสดงเข้ามา และแสดงผลดังนี้

```
Input number line : 5
Input character : @

@
@@
@@@
@@@@
@@@@@
```

3. ให้นักศึกษาเขียนโปรแกรม เพื่อตรวจสอบตัวอักษรในประโยคสตริงที่รับมาจากคีย์บอร์ดว่ามีตัวอักษรตัวใหญ่ ตัวเล็ก ตัวเลข ช่องว่าง(space) และสัญลักษณ์อื่นอยู่อย่างละเท่าไร โดยแสดงเฉพาะข้อมูลที่มีเท่านั้น ดังนี้

```
Enter message : This is a Good Time.
Your message have 3 uppers character.
Your message have 12 lowers character.
Your message have 4 spaces character.
Your message have 1 special character.
```