



Phần trình bày của:

**ĐẬU HẢI PHONG**

*Giảng viên*

*Đại Nam, ngày 01 tháng 19 năm 2023*

# LƯU Ý

**KHÔNG NÓI  
CHUYỆN RIÊNG**



**KHÔNG SỬ DỤNG  
ĐIỆN THOẠI**



**KHÔNG NGỦ GẬT**



**GHI CHÉP ĐẦY ĐỦ**





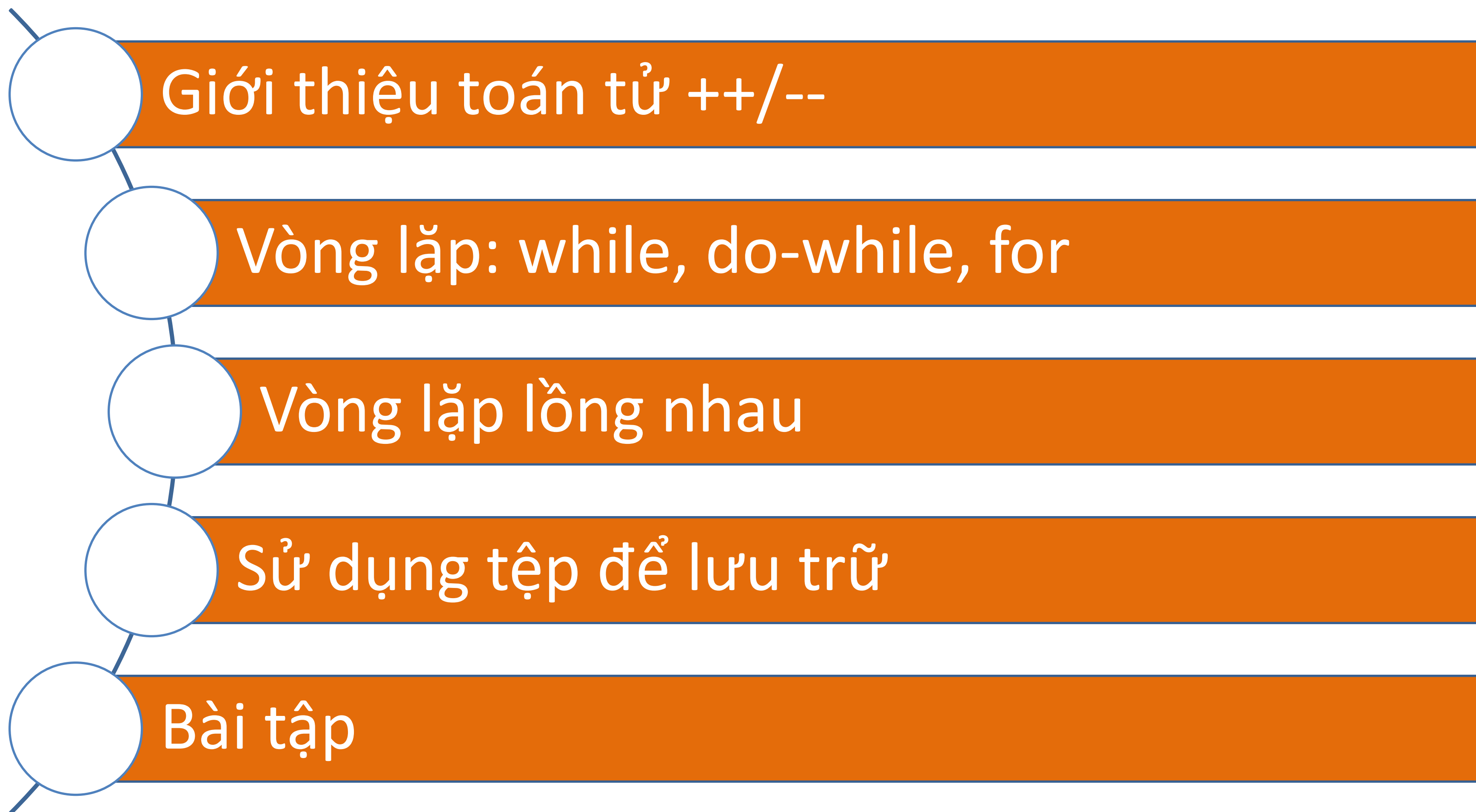
# LẬP TRÌNH CƠ BẢN



## CHƯƠNG 4

# CẤU TRÚC LẬP VÀ TẬP DỮ LIỆU





# Toán tử tăng/giảm

- ++ toán tử tăng, thêm 1
  - val++, tương đương với  $val = val + 1$
- -- toán tử giảm, giảm 1
  - val--, tương đương với  $val = val - 1$
- --, ++ có thể sử dụng trước (prefix) hoặc sau (posfix) biến
  - ++val      val++      --val      val--
  - VD1:  

```
int x = 10;  
cout <<x++; //In ra 10
```
  - VD2:  

```
int x = 10;  
cout <<++x; //In ra 11
```

# Toán tử tăng/giảm

## Program 5-1

```
1 // This program demonstrates the ++ and -- operators.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int num = 4;    // num starts out with 4.
8
9     // Display the value in num.
10    cout << "The variable num is " << num << endl;
11    cout << "I will now increment num.\n\n";
12
13    // Use postfix ++ to increment num.
14    num++;
15    cout << "Now the variable num is " << num << endl;
16    cout << "I will increment num again.\n\n";
17
18    // Use prefix ++ to increment num.
19    ++num;
20    cout << "Now the variable num is " << num << endl;
21    cout << "I will now decrement num.\n\n";
22
23    // Use postfix -- to decrement num.
24    num--;
25    cout << "Now the variable num is " << num << endl;
26    cout << "I will decrement num again.\n\n";
27
```

## Program 5-1 (continued)

```
28    // Use prefix -- to increment num.
29    --num;
30    cout << "Now the variable num is " << num << endl;
31    return 0;
32 }
```

## Program Output

The variable num is 4  
I will now increment num.

Now the variable num is 5  
I will increment num again.

Now the variable num is 6  
I will now decrement num.

Now the variable num is 5  
I will decrement num again.

Now the variable num is 4

# Toán tử tăng/giảm

- Có thể sử dụng trong biểu thức:
  - $result = ++num1 + --num2;$
  - $if (++num > limit)$
- Không thể thực hiện dạng biểu thức:

–  $result = (num1 + num2)++;$

```
#include <iostream>
using namespace std;
int main() {
    int result1,num1=1,num2=1;
    result1 = num1++ + --num2;
    cout<<result1<<endl;// In ra 1

    int result2,num3=1,num4=1;
    result2 = num3++ + num4--;
    cout<<result2<<endl;//In ra 2
}
```



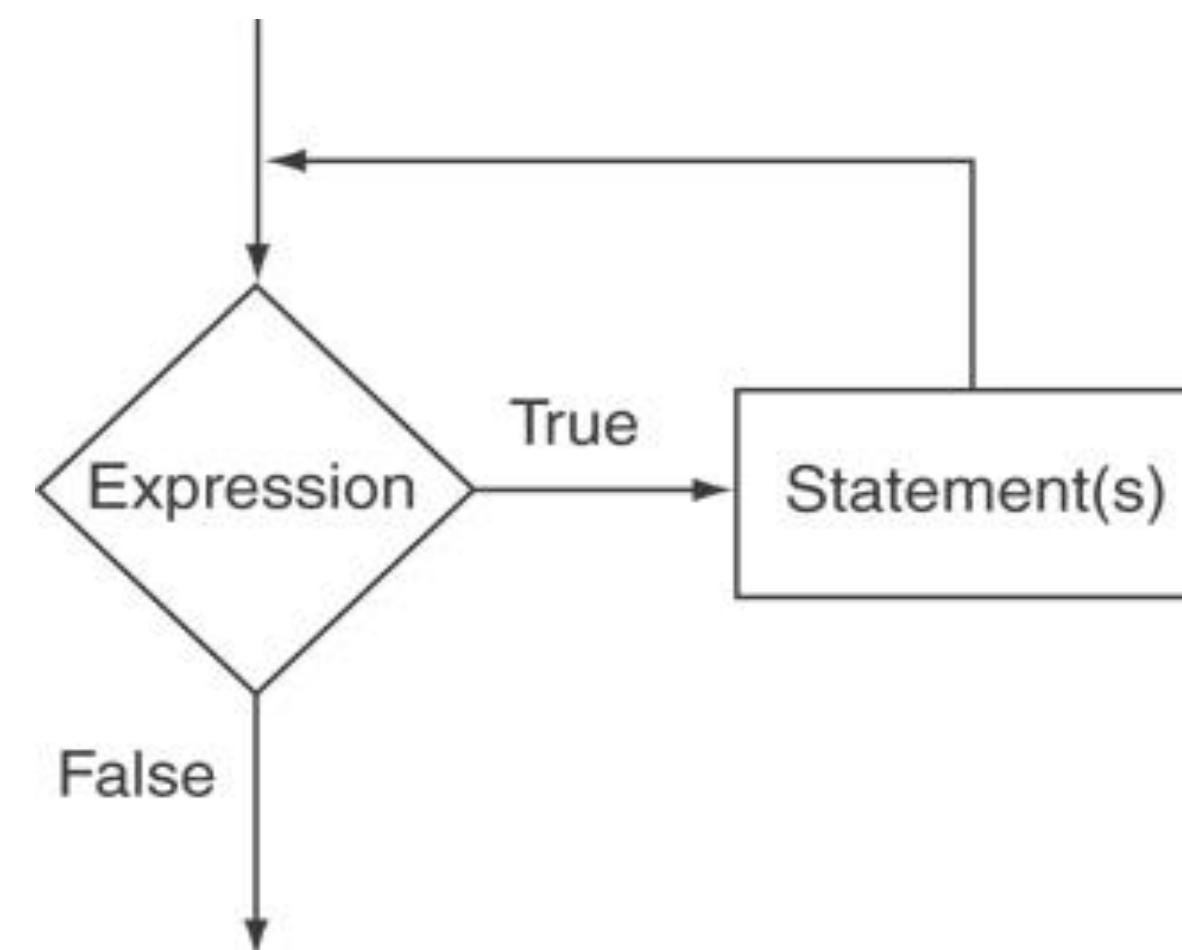
# Vòng lặp & vòng lặp while

- Vòng lặp là một cấu trúc kiểm soát hoặc cấu trúc lặp
- Cú pháp câu lệnh while:

while (<biểu thức>)

<câu lệnh>;

- Cũng có thể là nhiều câu lệnh được bao quanh {}
- <biểu thức> được kiểm tra:
  - Nếu đúng thì <câu lệnh> được thực hiện và <biểu thức> được kiểm tra lại.
  - Nếu sai, vòng lặp kết thúc và các câu lệnh dưới <câu lệnh> được thực hiện



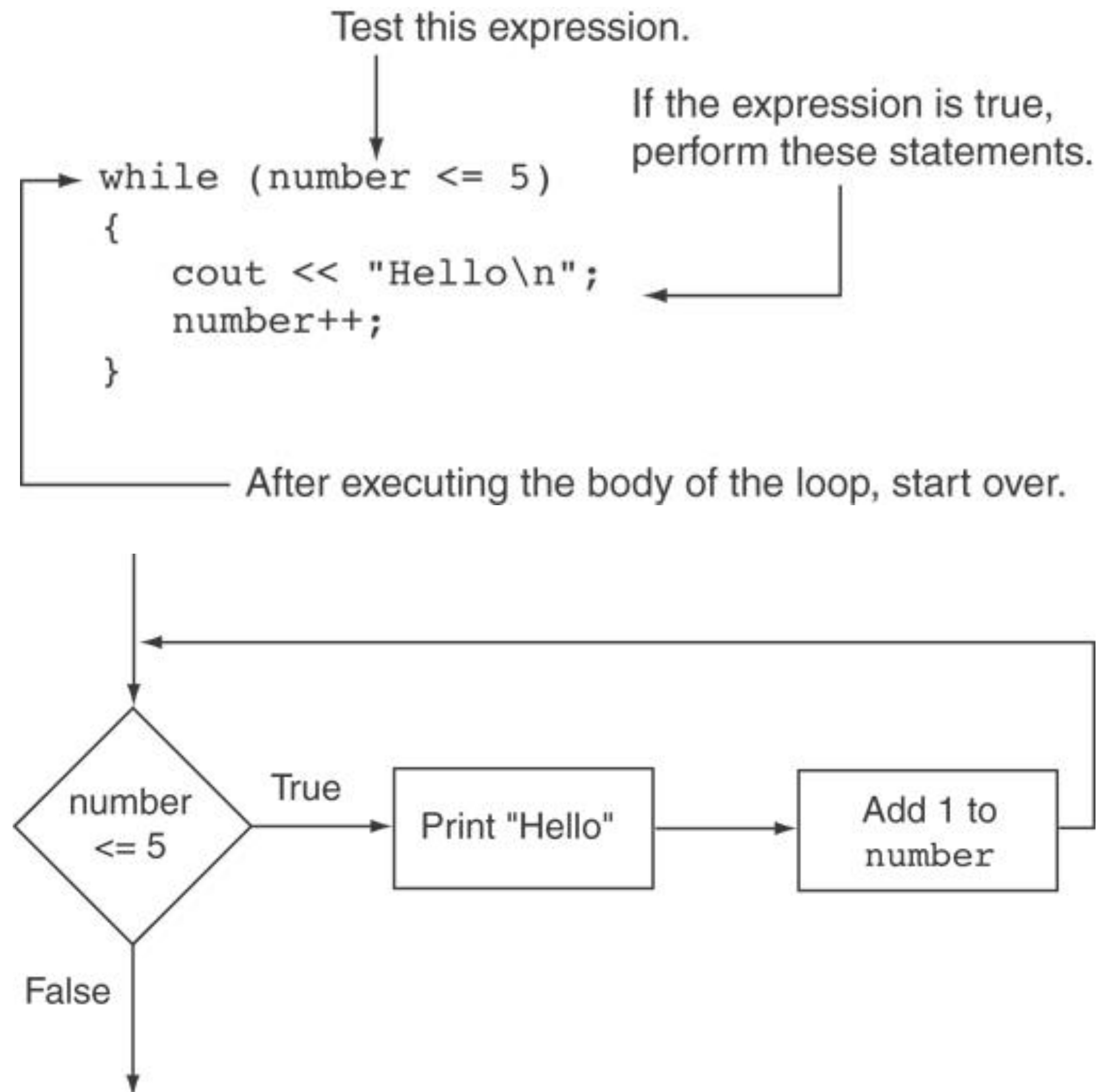
# Vòng lặp & vòng lặp while

## Program 5-3

```
1 // This program demonstrates a simple while loop.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int number = 1;
8
9     while (number <= 5)
10    {
11        cout << "Hello\n";
12        number++;
13    }
14    cout << "That's all!\n";
15    return 0;
16 }
```

## Program Output

```
Hello
Hello
Hello
Hello
Hello
That's all!
```



# Vòng lặp & vòng lặp while

- while là vòng lặp *kiểm tra điều kiện trước*
  - Đoạn mã sau sẽ không thực hiện. Tại sao?

```
int number = 6;
while (number <= 5)
{
    cout << "Hello\n";
    number++;
}
```

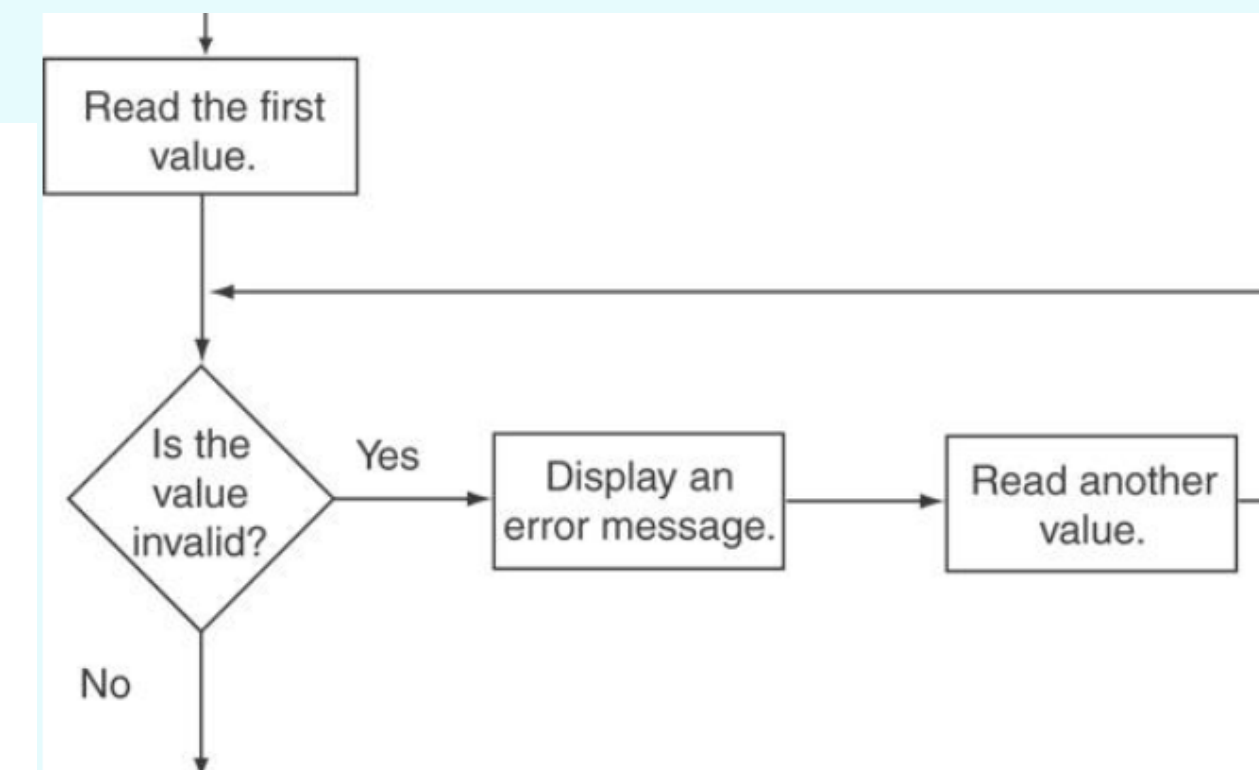
- Trong vòng lặp phải có đoạn mã làm cho <biểu thức> điều kiện lặp trở thành false. Nếu không sẽ là vòng lặp vô hạn
- VD:

```
int number = 1;
while (number <= 5)
{
    cout << "Hello\n";
}
```

# Xác thực đầu vào với while

- Xác thực là quá trình kiểm tra dữ liệu đầu vào xem có hợp lệ hay không?
- Câu lệnh while thường được dùng để từ chối dữ liệu không hợp lệ và lặp lại đến khi hợp lệ
- Giả mã:
  - Nhập 1 giá trị đầu vào
  - Trong khi (while) giá trị đầu vào sai
    - Thông báo lỗi
    - Nhập lại giá trị đầu vào
  - Kết thúc (while)

```
cout << "Enter a number less than 10: ";  
cin >> number;  
while (number >= 10)  
{  
    cout << "Invalid Entry!"  
        << "Enter a number less than 10: ";  
    cin >> number;  
}
```



# Kiểm tra

- Bài 1: Viết chương trình sử dụng vòng lặp để xác thực giá trị biến nhập vào trong phạm vi 10 đến 25.
- Bài 2: Viết chương trình sử dụng vòng lặp để xác thực giá trị đầu vào 'Y', 'y', 'N' hay 'n'.



# Biến đếm

- Biến đếm là 1 biến thường tăng lên hay giảm đi sau mỗi lần lặp.
- Thường dùng để kiểm soát và theo dõi số lần lặp.

- Ví dụ:

```
1 // This program has two loops. The first displays the numbers
2 // from 1 up to 5. The second displays the numbers from 5 down to 1.
3 // The program uses the ++ and -- operators to change the value
4 // of the loop control variables.
5 #include <iostream>
6 using namespace std;
7
8 int main()
9 {
10     int countUp = 1;    // Initialize the first loop control variable to 1
11     while (countUp < 6)
12     {
13         cout << countUp << " ";
14         countUp++;      // The ++ operator increments countUp
15     }
16     cout << endl << endl;
17
18     int countDown = 5;  // Initialize the second loop control variable to 5
19     while (countDown > 0)
20     {
21         cout << countDown << " ";
22         countDown--;    // The -- operator decrements countDown
23     }
24     return 0;
25 }
```



# Vòng lặp do-while

- Đây là vòng lặp được **kiểm tra điều kiện sau**
- Cú pháp:

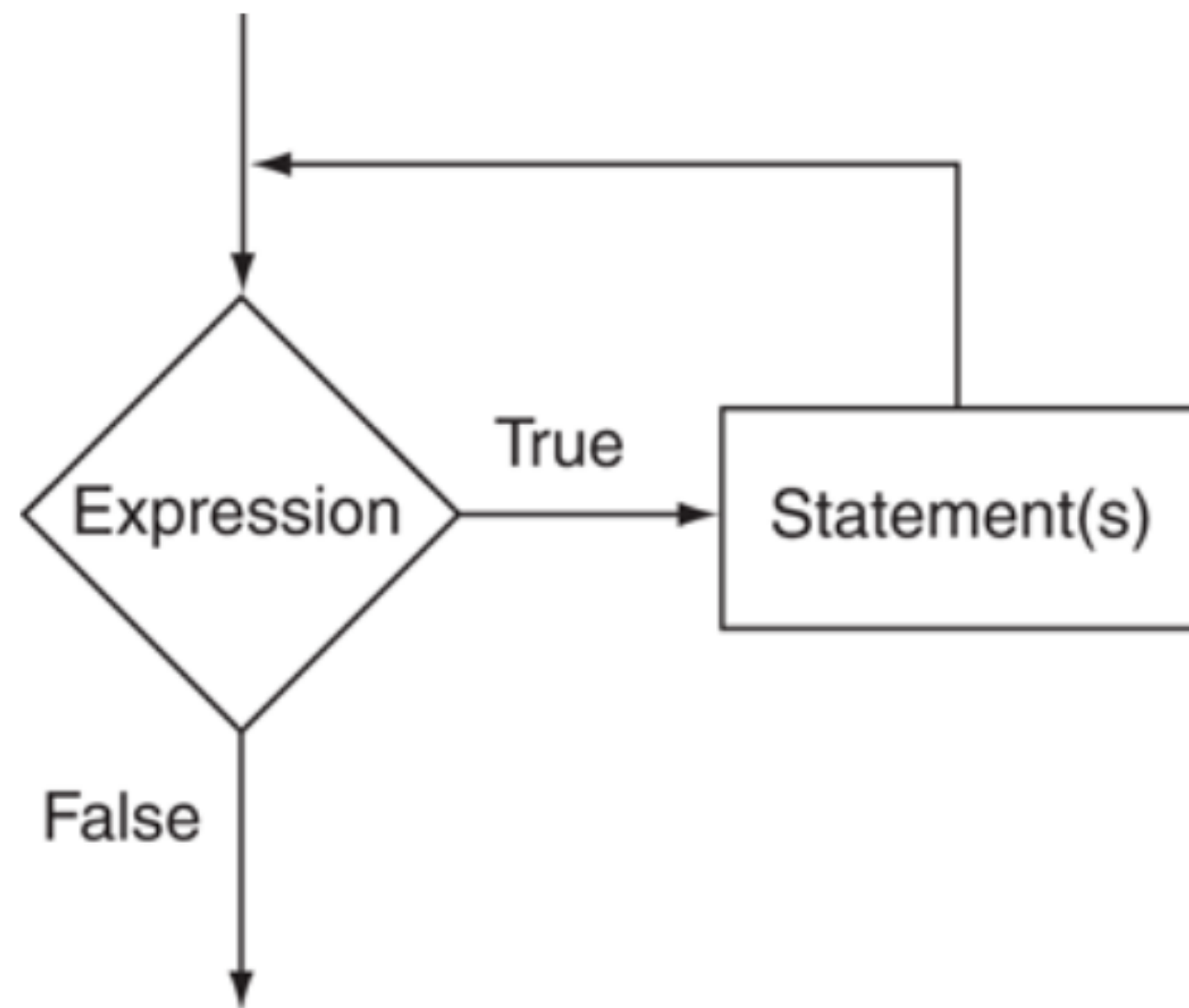
```
do  
    statement;  
while (expression);
```

```
do  
{  
    statement;  
    statement;  
    // Place as many statements here  
    // as necessary.  
} while (expression);
```

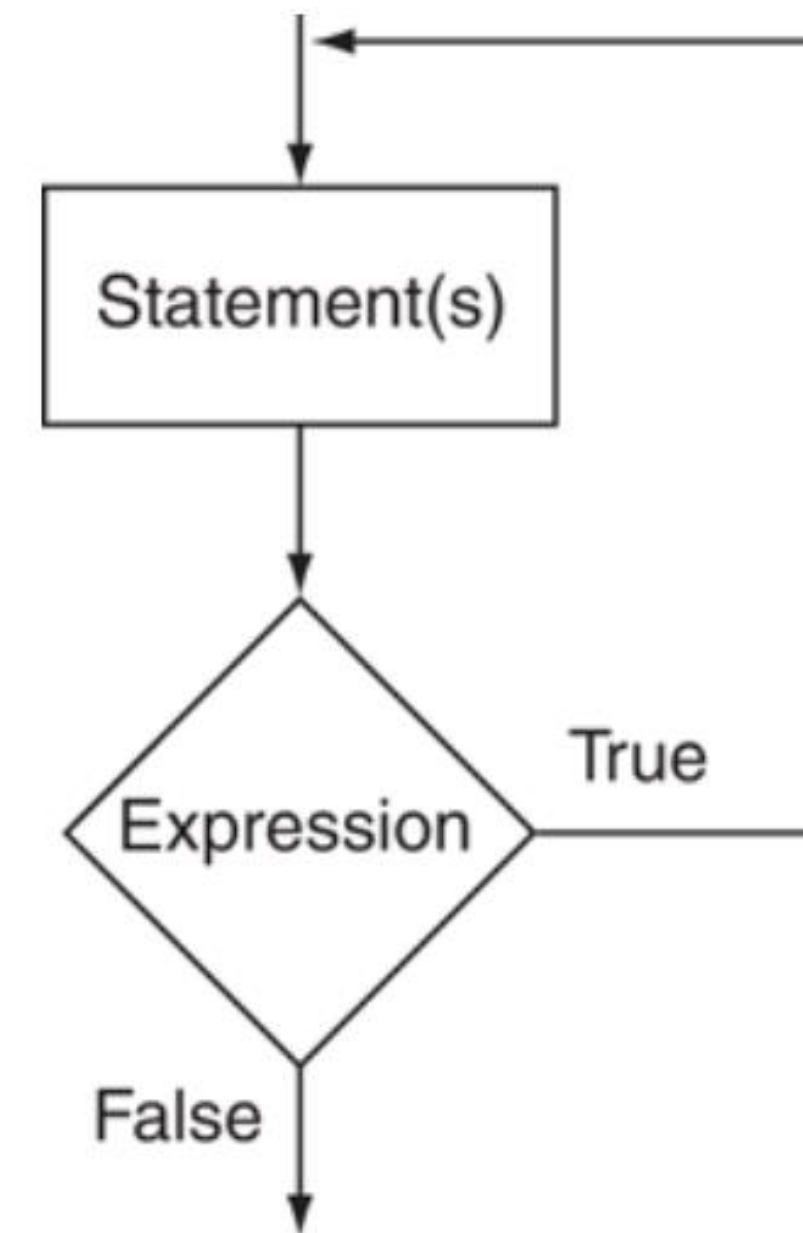
- Phải kết thúc bằng dấu **chấm phẩy “;”**
- Luôn thực hiện **ít nhất 1 lần lặp**.

# Vòng lặp do-while

- So sánh: *while* và *do-while*



```
int x = 1;
while (x < 0)
    cout << x << endl;
```

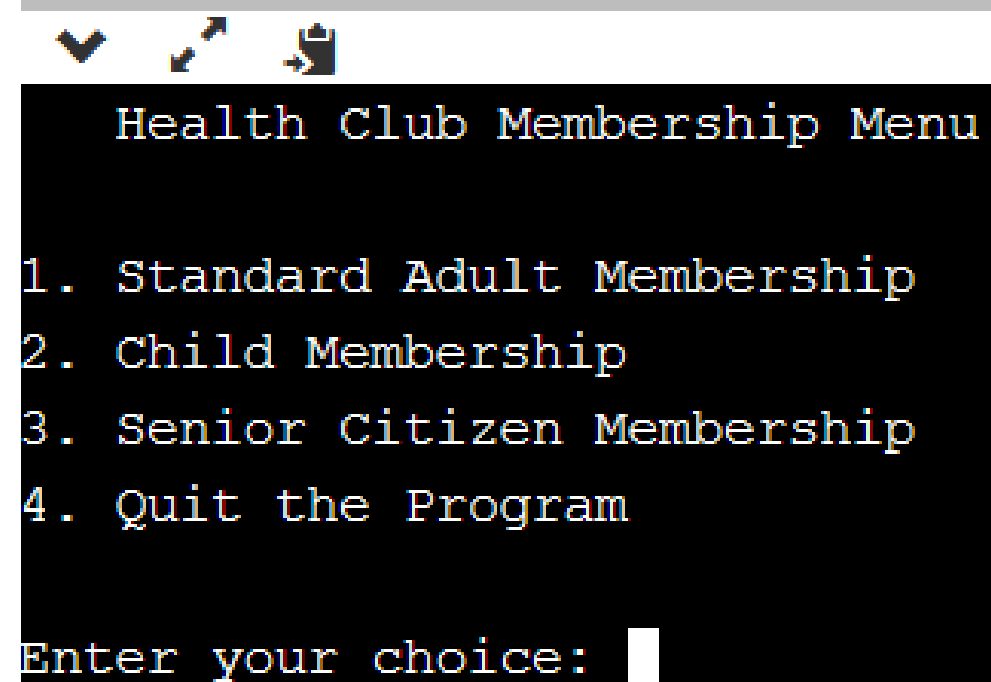


```
int x = 1;
do
    cout << x << endl;
while (x < 0);
```

# Sử dụng do-while với thực đơn

- Sửa lại Thực đơn của Program 4-27, để chương trình lặp lại đến khi người dùng chọn 4. (*Program 5-8*)

```
14     int choice;           // Menu choice
15     int months;          // Number of months
16     double charges;      // Monthly charges
17
18     // Display the menu and get the user's choice
19     cout << "    Health Club Membership Menu\n\n";
20     cout << "1. Standard Adult Membership\n";
21     cout << "2. Child Membership\n";
22     cout << "3. Senior Citizen Membership\n";
23     cout << "4. Quit the Program\n\n";
24     cout << "Enter your choice: ";
25     cin  >> choice;
26
```



```
Health Club Membership Menu

1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program

Enter your choice: █
```

# Vòng lặp for

- Thường dùng khi **biết trước số lần lặp**
- Cú pháp:

```
for (initialization; test; update)  
    statement;
```

```
for (initialization; test; update)  
{  
    statement;  
    statement;  
    // Place as many statements here  
    // as necessary.  
}
```

Bước 1: Thực hiện biểu thức khởi tạo

Bước 2, Kiểm tra biểu thức. Nếu biểu thức đúng thì thực hiện Bước 3. Ngược lại, thì kết thúc vòng lặp

```
for (count = 0; count < 5; count++)  
    cout << "Hello" << endl;
```

Bước 3, thực hiện các câu lệnh trong vòng lặp

Bước 4, thực hiện cập nhật biểu thức, sau đó quay lại Bước 2.

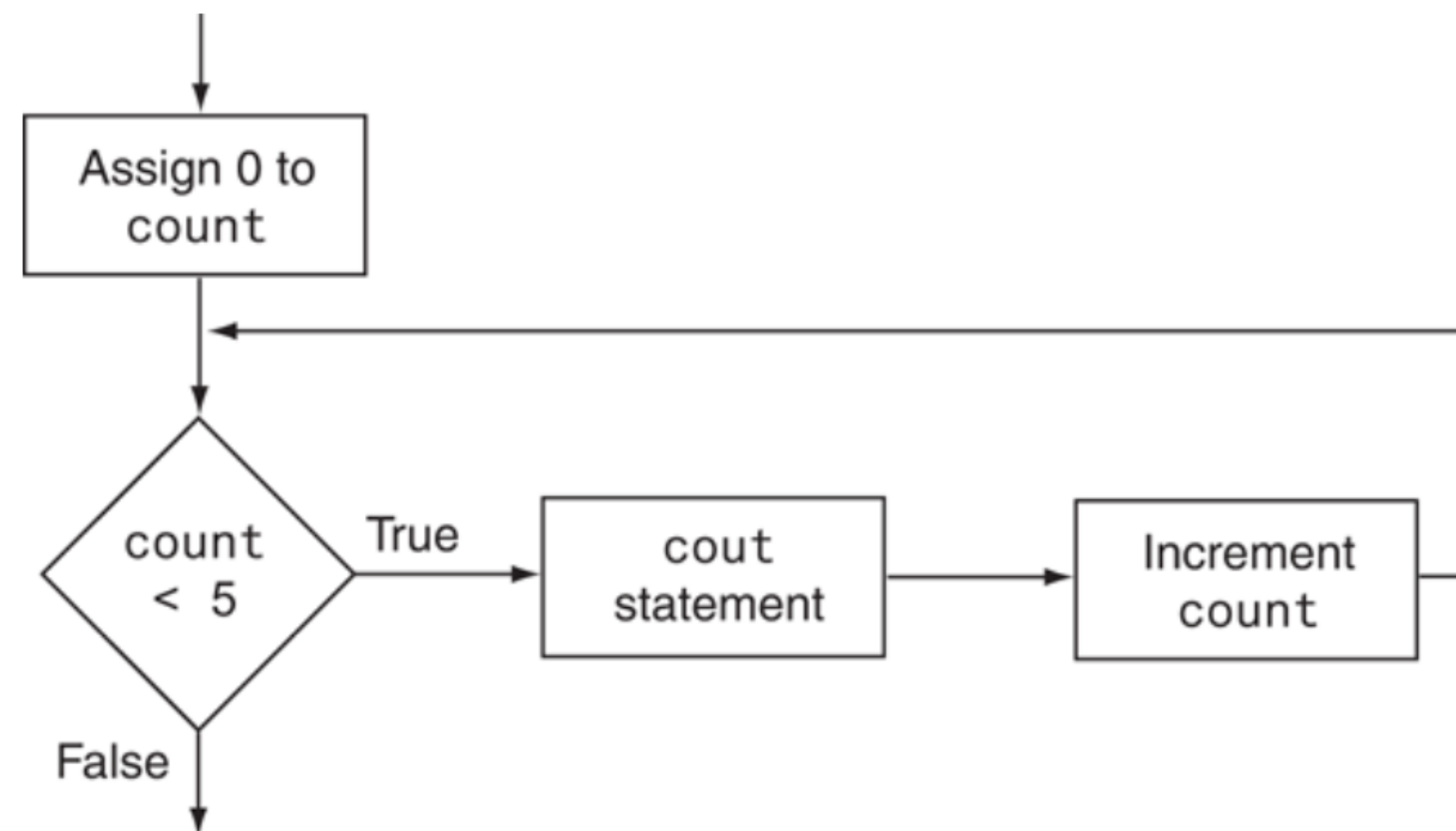
# Vòng lặp for

- Sơ đồ:

```
1 // This program uses a for loop to display the numbers 1-5
2 // and their squares.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 int main()
8 {   int num;
9
10     cout << "Number  Square\n";
11     cout << "-----\n";
12
13     for (num = 1; num <= 5; num++)
14         cout << setw(4) << num << setw(7) << (num * num) << endl;
15     return 0;
16 }
```

input

Number	Square
1	1
2	4
3	9
4	16
5	25



# Sử dụng for thay while hoặc do-while

- Ví dụ: sửa Program 5-6 thành vòng lặp for

Initialization expression

Test expression

```
int num = MIN_NUMBER;  
while (num <= MAX_NUMBER)  
{  
    cout << num << "\t\t" << (num * num) << endl;  
    num++;  
}
```

Update expression

Initialization expression

Test expression

Update expression

```
for (num = MIN_NUMBER; num <= MAX_NUMBER; num++)  
    cout << num << "\t\t" << (num * num) << endl;
```



# Một số ví dụ vòng lặp for

- Khai báo trong vòng lặp:

```
for (int num = minNumber; num <= maxNumber; num++)
```

- Khởi tạo nhiều giá trị:

```
int x, y;  
for (x = 1, y = 1; x <= 5; x++)  
{  
    cout << x << " plus " << y  
        << " equals " << (x + y)  
        << endl;  
}
```

- Bỏ qua khởi tạo:

```
int num = 1;  
for ( ; num <= maxValue; num++)  
    cout << num << "\t\t" << (num * num) << endl;
```

# Kiểm tra



- 5.6. Tên của 3 biểu thức bên trong dấu ngoặc đơn của vòng for là gì?
- 5.7. Giả sử bạn muốn viết 50 lần chữ “I love you” và sử dụng biến count.  
Vậy:
  - Biểu thức khởi tạo, kiểm tra và cập nhật sẽ như thế nào?
  - Viết vòng lặp for cho công việc trên.
- 5.8. Đoạn chương trình sau sẽ hiển thị gì?
  - A. `for (int count = 0; count < 6; count++) cout << (count + count);`
  - B. `for (int value = -5; value < 5; value++) cout << value;`
  - C. `for (int x = 5; x <= 14; x += 3) cout << x << endl;`
- 5.9. Viết chương trình nhập vào 7 số, sử dụng vòng lặp for, tính và in ra tổng 7 số vừa nhập.

# Vòng lặp lồng nhau

- Là vòng lặp trong 1 vòng lặp khác
- Giả sử đoạn mã sau in ra 00...59 giây

```
cout << fixed << right;  
cout.fill('0');  
for (int seconds = 0; seconds < 60; seconds++)  
    cout << setw(2) << seconds << endl;
```

- Hãy sửa lại đoạn mã trên để in ra 00:00...60:00
- Tiếp tục sửa đoạn mã trên để in ra 00:00:00 ...24:00:00
- Program 5-14: Viết chương trình, nhập vào số lượng SV và số môn SV phải học. Với từng sinh viên sẽ nhập điểm cho từng môn. Sau khi nhập xong điểm các môn của 1 sinh viên thì hiển thị điểm trung bình các môn của sinh viên đó.

# Tệp - File

- Gồm 2 loại tệp:
  - Văn bản:
    - Chứa các dữ liệu đã được mã hóa là văn bản (Mã ASCII hoặc Unicode)
    - Nếu là các con số thì các con số đó được lưu như 1 chuỗi kí tự
    - Có thể dùng chương trình soạn thảo mở để xem sửa (Notepad)
  - Nhị phân:
    - Chứa các dữ liệu chưa được chuyển đổi thành văn bản
    - Không thể mở để xem bằng trình soạn thảo văn bản
- Trong chương này chỉ làm việc với tệp **văn bản**

# Làm việc với tệp

- Có 2 cách truy cập tệp:
  - Tuần tự:
    - Nếu muốn truy cập vào dữ liệu ở cuối tệp tin thì phải đọc tất cả dữ liệu phía trước.
  - Trực tiếp:
    - Có thể nhảy đến 1 vài điểm của dữ liệu mà không cần đọc tuần tự.
    - Giống như đĩa CD, mp3,... nhảy trực tiếp đến 1 bài hát nào đó.
  - Trong chương này, chỉ tập trung vào truy cập tuần tự.
- Tệp trên ổ cứng được xác định bởi tên tệp
  - VD: cat.jpg, bai1.cpp, vanban.docx,...
- Đối tượng luồng tệp – File stream objects:
  - Làm việc rất giống với lệnh **cout** và **cin**
  - Một luồng dữ liệu dữ liệu được gửi đến đối tượng luồng tệp để ghi vào tệp.
  - Dữ liệu được đọc từ file, dữ liệu từ đối tượng luồng tệp kết hợp với tệp để đưa vào các biến

# Làm việc với tệp

- Thiết lập chương trình có I/O với tệp:
  - Để làm việc với Vào/Ra tệp cần có **#include <fstream>**
- Tạo đối tượng tệp và mở tệp:
  - Trước khi **ghi** hoặc **đọc** từ tệp thì:
    - Một đối tượng tệp phải được tạo
    - Tệp phải được **mở** và liên kết với đối tượng tệp (Tên tệp)
  - VD1: Mở 1 tệp để đọc dữ liệu từ tệp:

```
ifstream inputFile; //Tạo 1 đối tượng để tên inputFile để đọc từ tệp  
inputFile.open("Customers.txt"); //Mở tệp Customers.txt liên kết với inputFile
```

- VD2: Mở 1 tệp để ghi dữ liệu vào tệp:

```
ofstream outputFile; //Tạo 1 đối tượng để tên outputFile để ghi vào tệp  
outputFile.open("Employees.txt"); //Mở tệp Employees.txt liên kết với outputFile
```

- Nếu tên tệp không nằm trong thư của tệp chương trình thì cần chỉ rõ
  - VD: `inputFile.open("C:\\data\\inventory.txt")`



# Làm việc với tệp

- Đóng tệp:
  - Nên đóng tệp sau khi Đọc/Ghi tệp vì:
    - Có thể có những dữ liệu trong bộ nhớ đệm chưa được ghi vào tệp
    - Giới hạn số lượng tệp được mở.
  - VD: đóng tệp gắn với đối tượng inputFile  
`inputFile.close();`
- Ghi dữ liệu vào tệp:  
`outputFile << "I love C++ programming\n";`  
`outputFile << "Price: " << price << endl;`

# Làm việc với tệp

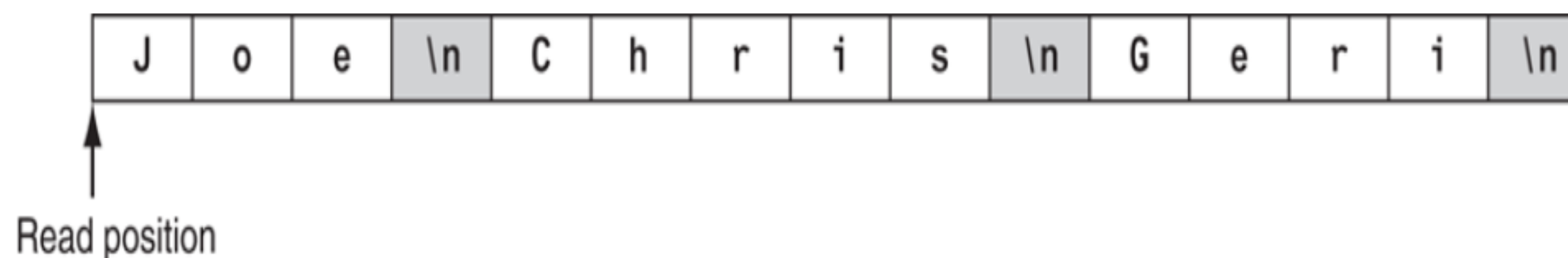
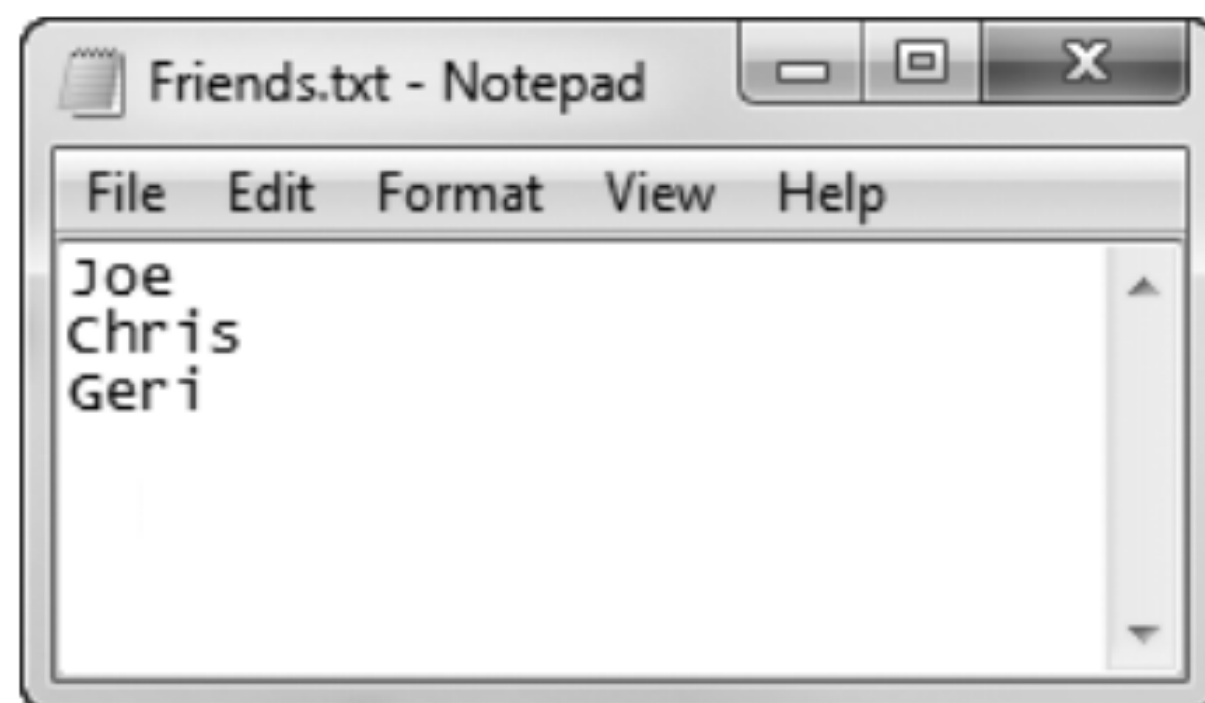
pr5-16.cpp

```
1 // This program writes data to a file.
2 #include <iostream>
3 #include <fstream>          // Needed to use files
4 using namespace std;
5
6 int main()
7 {
8     ofstream outputFile;
9
10    // Open the output file
11    outputFile.open("demofile.txt");
12
13    cout << "Now writing data to the file.\n";
14
15    // Write four names to the file
16    outputFile << "Bach\n";
17    outputFile << "Beethoven\n";
18    outputFile << "Mozart\n";
19    outputFile << "Schubert\n";
20
21    // Close the file
22    outputFile.close();
23
24    cout << "Done.\n";
25    return 0;
26 }
27
```

```
1 // This program writes user input to a file.
2 #include <iostream>
3 #include <fstream>          // Needed to use files
4 #include <string>
5 using namespace std;
6
7 int main()
8 {
9     ofstream outputFile;
10    string name1, name2, name3;
11
12    // Open the output file
13    outputFile.open("Friends.txt");
14
15    // Get the names of three friends
16    cout << "Enter the names of three friends.\n";
17    cout << "Friend #1: ";
18    cin >> name1;
19    cout << "Friend #2: ";
20    cin >> name2;
21    cout << "Friend #3: ";
22    cin >> name3;
23
24    // Write the names to the file
25    outputFile << name1 << endl;
26    outputFile << name2 << endl;
27    outputFile << name3 << endl;
28
29    // Close the file
30    outputFile.close();
31
32    cout << "The names were saved to a file.\n";
33    return 0;
34 }
```

# Vị trí đọc tệp

- Khi 1 tệp được mở thì đối tượng luồng tệp duy trì vị trí đọc tệp
- Vị trí đọc tệp xác định giá trị (byte) tiếp theo sẽ được đọc.
- Khi 1 tệp đầu vào được mở vị trí đọc của tệp là byte **đầu tiên**
- Vị trí đọc sẽ di chuyển về cuối tệp tin
- Giả sử có tệp Friends.txt như sau:

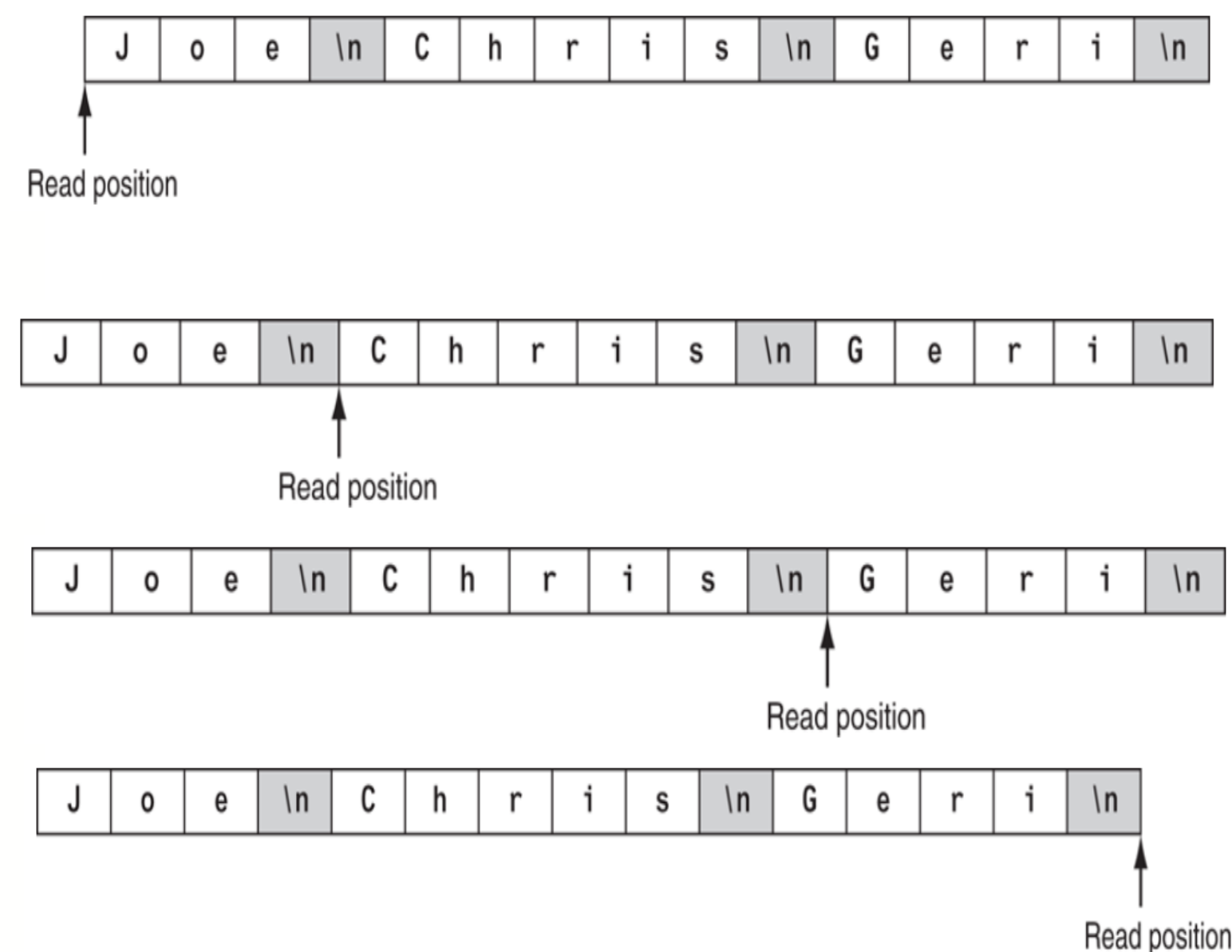


# Vị trí đọc tệp

```

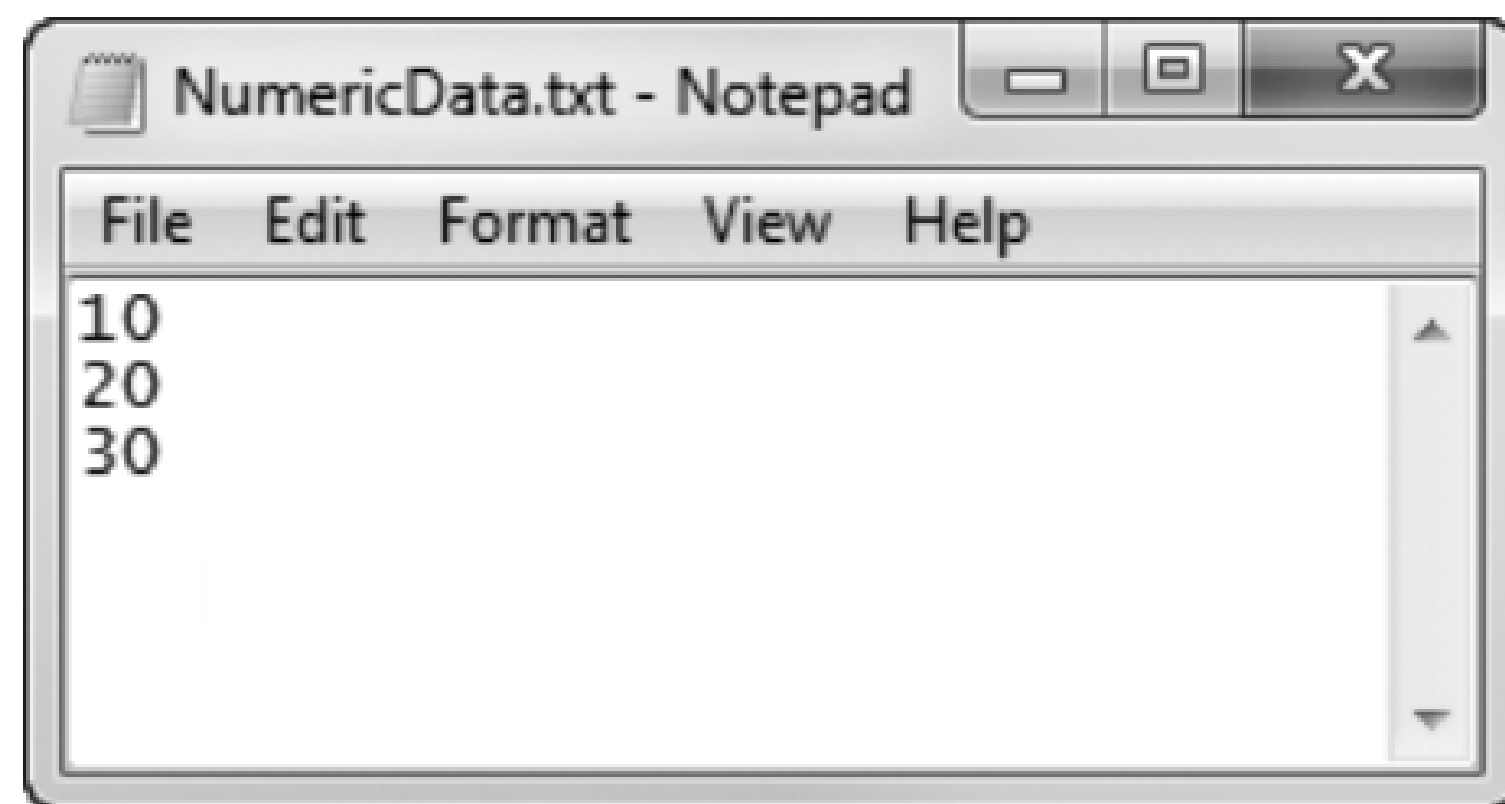
1 // This program reads data from a file.
2 #include <iostream>
3 #include <fstream>           // Needed to use files
4 #include <string>
5 using namespace std;
6
7 int main()
8 {
9     ifstream inputFile;
10    string name;
11
12    // Open the input file
13    inputFile.open("Friends.txt");
14
15    cout << "Reading data from the file.\n";
16
17    inputFile >> name;           // Read name 1 from the file and display it
18    cout << name << endl;
19
20    inputFile >> name;           // Read name 2 from the file and display it
21    cout << name << endl;
22
23    inputFile >> name;           // Read name 3 from the file and display it
24    cout << name << endl;
25
26    inputFile.close();           // Close the file
27    return 0;
28 }

```



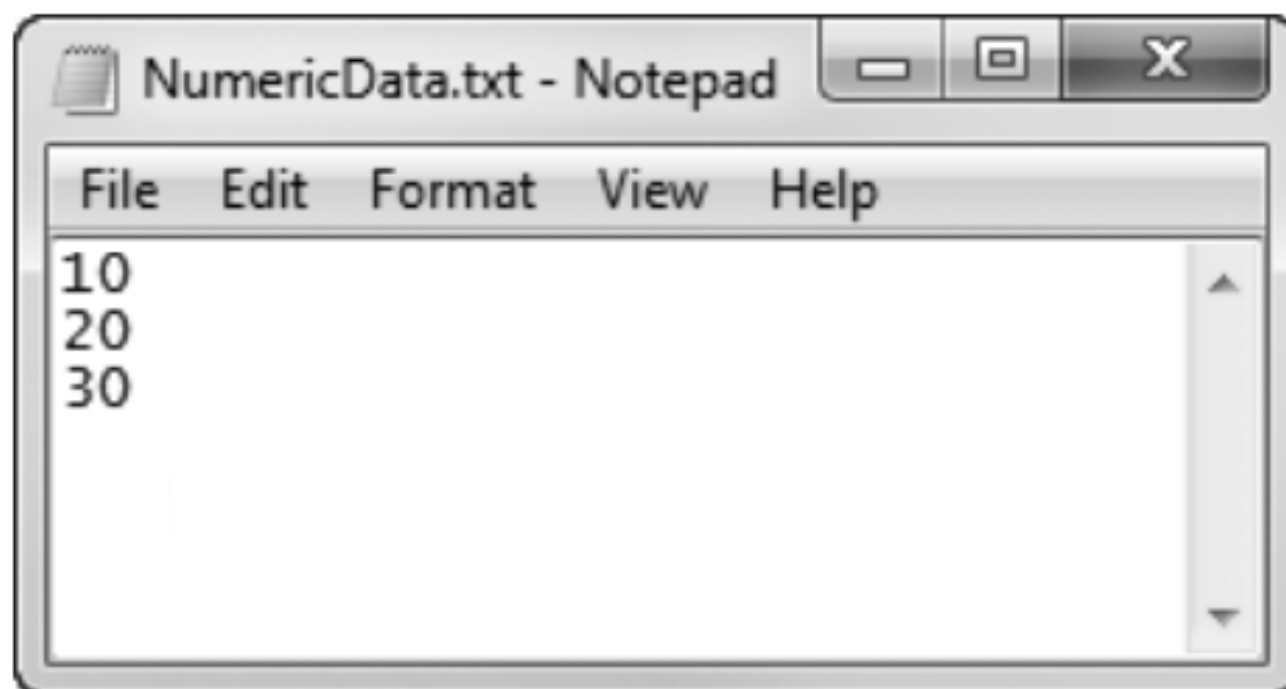
# Đọc dữ liệu số từ tệp văn bản

- Dữ liệu được lưu trong tệp văn bản thì luôn được mã hóa dạng văn bản sử dụng mã ASCII hoặc Unicode.
- Nếu là số thì cũng cũng là chuỗi kí tự. VD: “123”.
- Sử dụng toán tử >> để đọc dữ liệu dạng số ra biến số.
- Toán tử >> **tự động chuyển đổi thành dữ liệu kiểu số**



# Đọc dữ liệu số từ tệp văn bản

- Program 5-20:
  - Tệp đầu vào:



- Kết quả

```
Here are the numbers:
10 20 30
Their sum is: 60
```

```
1 // This program reads numbers from a file.
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5
6 int main()
7 {
8     ifstream inFile;
9     int value1, value2, value3, sum;
10
11     // Open the file.
12     inFile.open("NumericData.txt");
13
14     // Read the three numbers from the file.
15     inFile >> value1;
16     inFile >> value2;
17     inFile >> value3;
18
19     // Close the file.
20     inFile.close();
21
22     // Calculate the sum of the numbers.
23     sum = value1 + value2 + value3;
24
25     // Display the three numbers.
26     cout << "Here are the numbers:\n" << value1 << " " << value2 << " " << value3 << endl;
27
28     // Display the sum of the numbers.
29     cout << "Their sum is: " << sum << endl;
30     return 0;
31 }
```

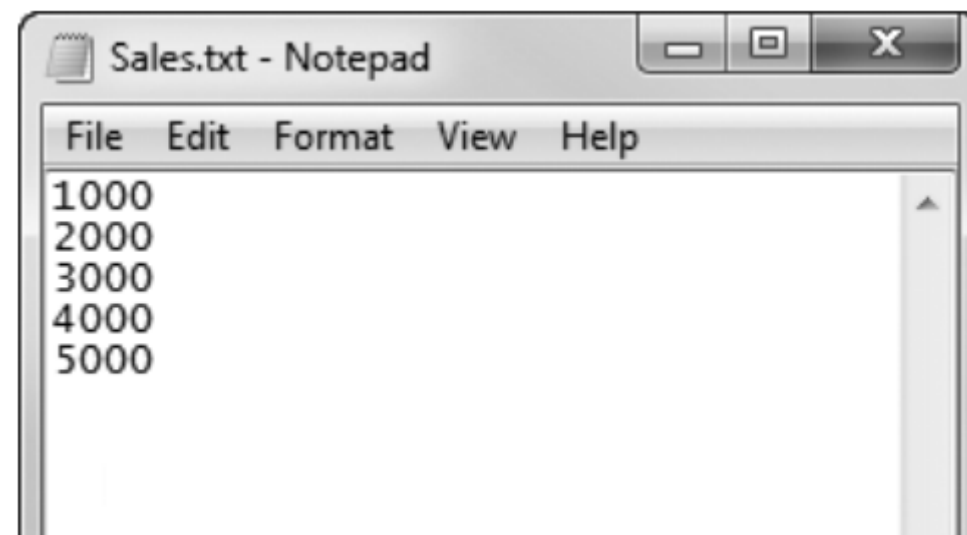


# Sử dụng vòng lặp để xử lý tệp

- Các tệp tin thường sử dụng lưu lượng lớn dữ liệu
- VD: Dùng vòng lặp để nhập dữ liệu và ghi dữ liệu vào tệp

```
For how many days do you have sales? 4
Enter the sales for day 1: 1000
Enter the sales for day 2: 2000
Enter the sales for day 3: 3000
Enter the sales for day 4: 4000
Data written to Sales.txt
```

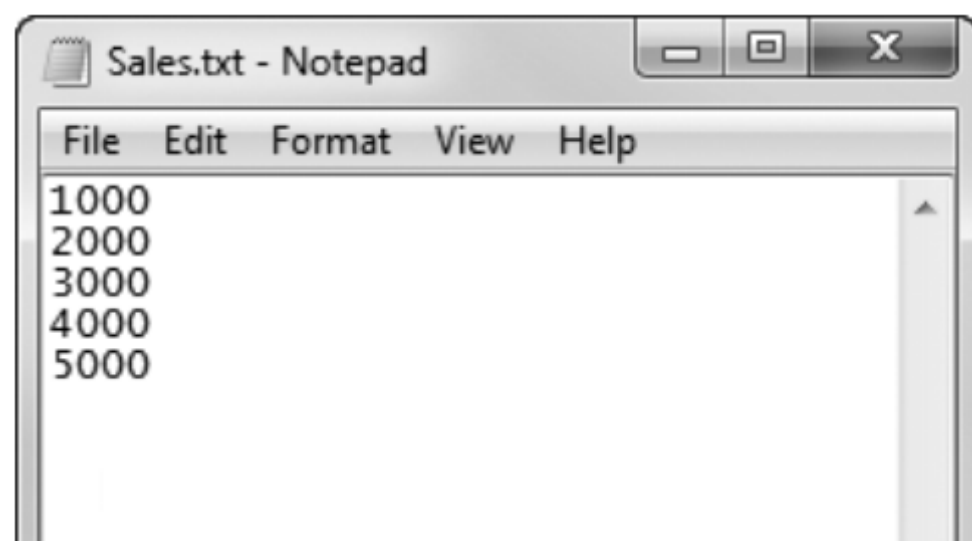
— Kết quả: tạo tệp Sales.txt



```
1 // This program uses a loop to write multiple values to a file.
2 #include <iostream>
3 #include <fstream> // Needed to use files
4 using namespace std;
5
6 int main()
7 {
8     ofstream outputFile; // File stream object
9     int numberOfDays; // Number of days of sales
10    double sales; // Sales amount for a day
11
12    // Get the number of days
13    cout << "For how many days do you have sales? ";
14    cin >> numberOfDays;
15
16    // Open the output file
17    outputFile.open("Sales.txt");
18
19    // Loop once for each day of sales
20    for (int count = 1; count <= numberOfDays; count++)
21    {
22        // Get the sales amount for a day
23        cout << "Enter the sales for day " << count << ": ";
24        cin >> sales;
25
26        // Write the sales amount to the file
27        outputFile << sales << endl;
28    }
29    // Close the file
30    outputFile.close();
31
32    cout << "Data written to Sales.txt\n";
33    return 0;
34 }
```

# Sử dụng vòng lặp để xử lý tệp

- Các tệp tin thường sử dụng lưu lượng lớn dữ liệu
- VD: Sử dụng vòng lặp để đọc dữ liệu từ tệp.



```
How many days of sales data are stored in your file? 2
Total sales for the 2 days were $3000>
```

```
How many days of sales data are stored in your file? 3
Total sales for the 3 days were $6000>
```

```
How many days of sales data are stored in your file? 4
Total sales for the 4 days were $10000>
```

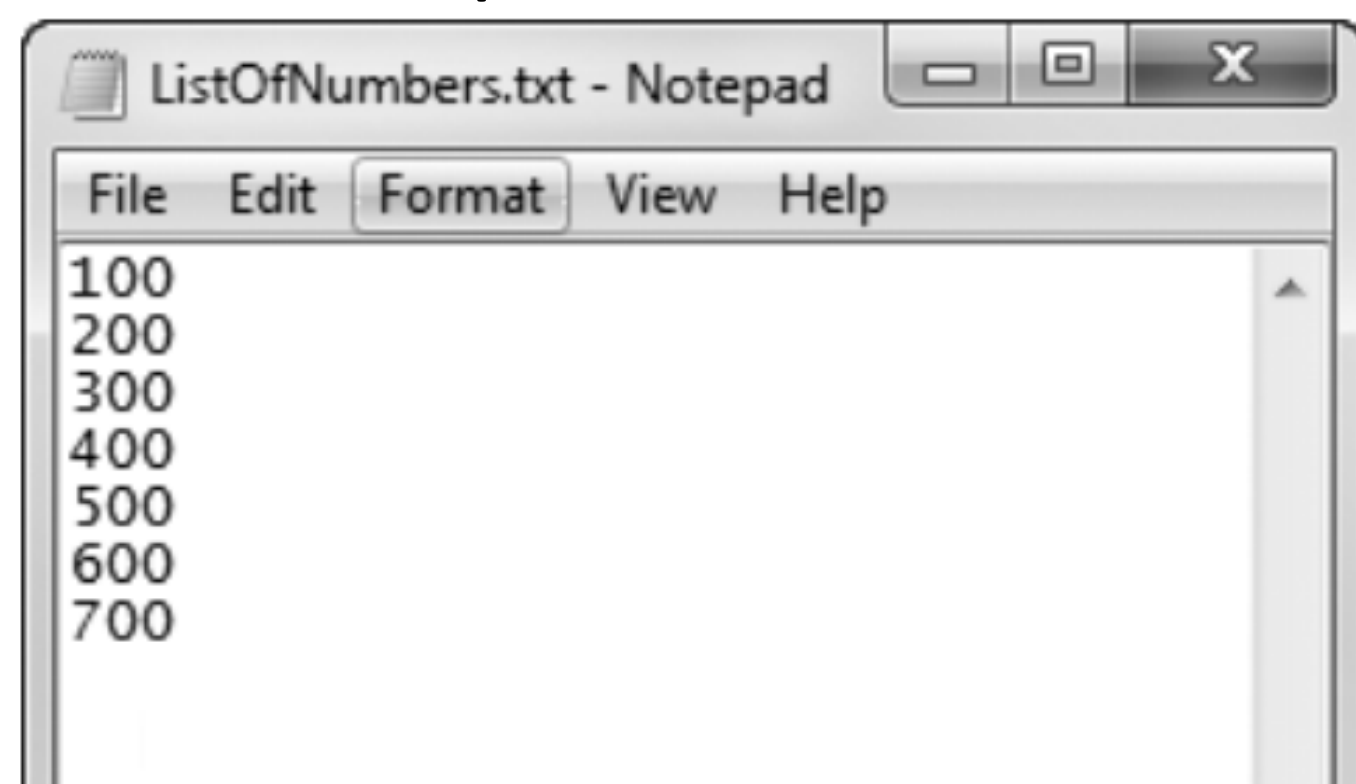
```
How many days of sales data are stored in your file? 5
Total sales for the 5 days were $14000>
```

```
1 // This program uses a loop to read in multiple values from a file.
2 #include <iostream>
3 #include <fstream> // Needed to use files
4 using namespace std;
5
6 int main()
7 {
8     ifstream inputFile; // File stream object
9     int numberOfDays; // Number of days of sales
10    double sales, // Sales amount for a day
11        totalSales = 0.0; // Accumulator
12
13    // Get the number of days
14    cout << "How many days of sales data are stored in your file? ";
15    cin >> numberOfDays;
16
17    // Open the input file
18    inputFile.open("Sales.txt");
19
20    // Loop once for each piece of data to be read from the file
21    for (int count = 1; count <= numberOfDays; count++)
22    {
23        // Read a sales figure from the file and add it to the sum
24        inputFile >> sales;
25        totalSales += sales;
26    }
27    // Close the file
28    inputFile.close();
29
30    cout << "Total sales for the " << numberOfDays << " days were $"
31        << totalSales;
32
33    return 0;
34 }
```

# Xác định kết thúc tệp

- Giả sử cần đọc và hiển thị tất cả nội dung của tệp. Nhưng không biết số lượng phần tử trong tệp.
- Có thể lỗi xảy ra khi đọc vượt quá phần tử cuối.
- Toán tử >> trả về Đúng nếu đọc thành công, ngược lại trả về Sai.

— Tệp đầu vào:



Kết quả:

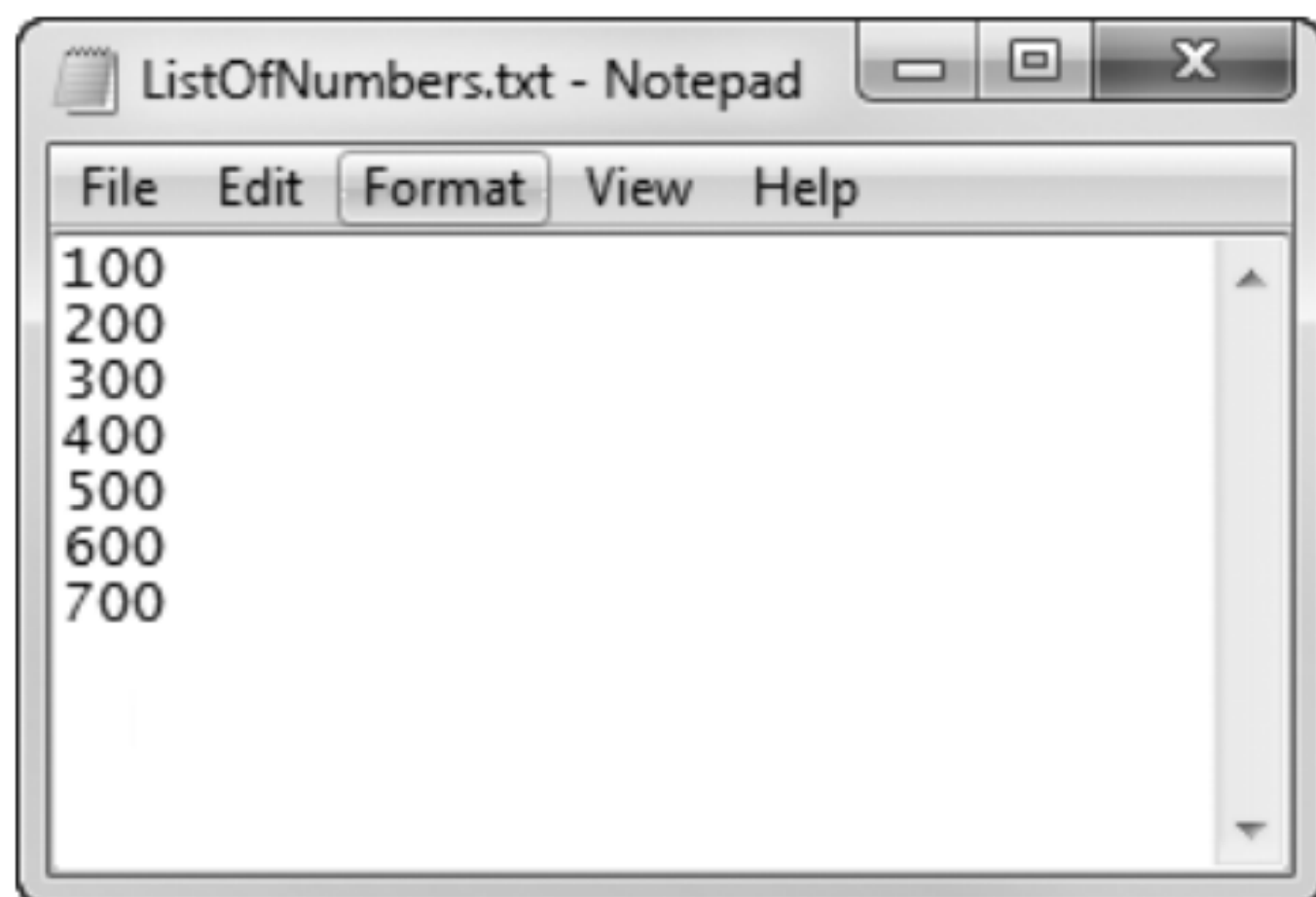
100  
200  
300  
400  
500  
600  
700

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      ifstream inputFile;
8      int number;
9
10     // Attempt to open the input file
11     inputFile.open("ListOfNumbers.txt");
12
13     // Read the numbers from the file and display them
14     while (inputFile >> number)
15         cout << number << endl;
16
17     // Close the file
18     inputFile.close();
19     return 0;
20 }
```

# Kiểm tra mở tệp

- Đối tượng luồng tệp trả về Đúng nếu mở thành công, ngược lại trả về Sai.

– Tệp đầu vào:



Kết quả:

100  
200  
300  
400  
500  
600  
700

```
1 // This program tests for file open errors.
2 #include <iostream>
3 #include <fstream> // Needed to use files
4 using namespace std;
5
6 int main()
7 {
8     ifstream inputFile;
9     int number;
10
11     // Attempt to open the input file
12     inputFile.open("ListOfNumbers.txt");
13
14     // If the file successfully opened, process it
15     if (inputFile)
16     {
17         // Read the numbers from the file and display them
18         while (inputFile >> number)
19             cout << number << endl;
20
21         // Close the file
22         inputFile.close();
23     }
24     else // The file could not be found and opened
25     {
26         // Display an error message
27         cout << "Error opening the file.\n";
28     }
29     return 0;
30 }
31
```

# Người dùng nhập vào tên tệp

- Program 5-23: Chương trình cho phép người dùng nhập vào tên tệp

```
7  int main()
8  {
9      ifstream inputFile;
10     string filename;
11     int number;
12
13     // Get the filename from the user
14     cout << "Enter the filename: ";
15     cin >> filename;
16     // Open the input file
17     inputFile.open(filename.c_str());
18
19     // If the file successfully opened, process it
20     if (inputFile)
21     {
22         // Read the numbers from the file and display them
23         while (inputFile >> number)
24             cout << number << endl;
25
26         // Close the file.
27         inputFile.close();
28     }
29     else
30     {
31         // Display an error message
32         cout << "Error opening the file.\n";
33     }
34     return 0;
35 }
36
```

# Kiểm tra



- 5.16. Tập đầu ra là gì? Tập đầu vào là gì?
- 5.17. Có 3 bước cần có khi 1 tệp được sử dụng?
- 5.18. Sự khác biệt giữa tệp văn bản và tệp nhị phân?
- 5.19. Sự khác biệt giữa truy cập tuần tự và truy cập ngẫu nhiên là gì?
- 5.20. Kiểu đối tượng luồng tệp gì khi muốn ghi dữ liệu vào tệp?
- 5.21. Kiểu đối tượng luồng tệp gì khi muốn đọc dữ liệu vào tệp?
- 5.22. Viết 1 chương trình, sử dụng vòng lặp for để ghi từ 1 đến 10 vào tệp.
- 5.23. Viết chương trình để đọc các dữ liệu được tạo ra từ 5.22.
- *Sinh viên tham khảo thêm Program: 5-25, 5-26 trang 913*



# HỎI ĐÁP



# Bài tập

- **Bài tập chương 4:**
  - Chapter4\_Programming Challenges.pdf



**Trân trọng cảm ơn!**

**ĐẬU HẢI PHONG**

*Giảng viên*

*dauhaiphong@dainam.edu.vn*

*0912441435*