บทที่ 8

Streams and Files

วัตถุประสงค์

1. เพื่อให้นักศึกษาทราบถึง streams และ แฟ้มข้อมูลของภาษา C++
2. เพื่อให้นักศึกษาทราบถึงไลบรารี่ที่สนับสนุน stream input และ output
3. เพื่อให้นักศึกษาทราบ เข้าใจ และสามารถประยุกต์ใช้แฟ้มในการแก้ปัญหาโปรแกรมได้

ในบทนี้เราจะกล่าวถึง คำสั่งที่สนับสนุน Stream ของ input และ output การจัดการแฟ้มข้อมูล ในดิสก์

8.1 The Standard Input/Output Streams

Streams เป็นลำดับของข้อมูลที่นำเข้าหรือแสดงผลในโปรแกรม มาตรฐานของ input Stream มีชื่อว่า cin และมาตรฐานของ output Stream มีชื่อว่า cout ซึ่งถูกนำมาเก็บรวมกันใน Library ที่ชื่อว่า iostream โดย cin จะเป็นการเชื่อมต่อกับแป้นพิมพ์ เมื่อโปรแกรมนั้นมีการอ่านอักขระ ในกรณีของ cout จะเป็นการเชื่อมต่อกับจอแสดงผล เมื่อมีการพิมพ์ตัวอักษรออกทางจอแสดงผล ในกรณีการ cin จะมีเบื้องหลังการทำงานคือในระหว่างที่ป้อนข้อมูลตัวกระทำที่เรียกว่า Stream Extraction operator (>>) จะทำการแปลงลำดับของข้อมูลให้อยู่ในรูปแบบที่เหมาะสมที่สามารถเก็บในหน่วยความจำได้อย่างถูกต้อง สมมุติว่ารับข้อมูลจากแป้นพิมพ์เป็น 123.45 ตัวกระทำ(>>) จะแปลงลำดับของตัวอักขระ ให้อยู่ในรูปแบบของเลขฐานสอง (Binary code) เก็บในหน่วยความจำ กรณีของการแสดงผลเราใช้ตัวกระทำที่เรียกว่า Insertion operator (<<) ในการแปลงค่าที่เก็บภายในหน่วยความจำให้เป็นลำดับของตัวอักษรเพื่อแสดงออกทางจอแสดงผล

ตัวอย่างที่ 8.1 การแปลงข้อมูลของ Input/output Stream

Input conversion

```
cin >> month >> day >>year;
```

cin (stream of characters entered at the keyboard)

เมื่อเราใช้คำสั่ง cin ระบบจะอ่านข้อมูลที่เราป้อนเข้าสู่ระบบเป็นสายของตัวอักขระถ้าผู้ใช้ป้อนข้อมูลเป็นตัวเลข 3 จำนวนโดยใช้ช่องว่างเป็นตัวแยกข้อมูลดังนี้

1	2		0	9		4	2	

ระบบจะตรวจสอบชนิดของข้อมูลในกรณีที่เป็นตัวเลขจำนวนเต็มจะอ่านทีละจำนวนมีผลให้ month มีค่าเท่ากับ 12 day มีค่าเท่ากับ 09 และ year มีค่าเท่ากับ 42

Output conversion (from internal representation to characters)

cout << "Enter " << setw (3) << n << " float values separated by a blank."

เราใช้คำสั่ง cout ในการแสดงสายของข้อมูลออกทางจอภาพ ตามที่เรากำหนดจากคำสั่งข้างต้น สมมุติว่า n เก็บค่า 203 ไว้ในหน่วยความจำ ระบบจะพิมพ์ข้อความเป็นลำดับดังนี้

E	n	t	e	r		2	0	3		f	l	o	a	t		v	...

ตัวอย่างที่ 8.2 โปรแกรม countChars

โปรแกรมต่อไปนี้เป็นการนับและแสดงผลจำนวนของตัวอักขระในแต่ละบรรทัดที่ป้อนทางแป้นพิมพ์

```
//File: countChars.cpp
//Counts numbers of characters in a line and lines in a stream

#include <iostream>
#include<string>
using namespace std;
#define  ENDFILE "CTRL-Z"
int  main()
{
        const char NWLN = '\n' ;    // newline character
        char next;                  // next character in current line
        int charCount;              // number of characters in current line
        int lineCount;              // keeps track of number of lines in file
```

```
    lineCount = 0;
    cout<<"Enter a line or press" << ENDFILE << ":";
    cin.get (next);                 // get first char of new line
    while (!cin.eof())
    {
        charCount = 0;      // initialize character count for new line
        while (next  !=  NWLN)
        {
            cout.put(next);
            charCount++;      //Increment character count.
            cin.get(next);      //Get next character.
        } // end inner while

    cout.put (NWLN);                    // Mark end of current display line.
    lineCount++;
    cout << "Number of characters in line "<<lineCount
        << " is "  << charCount<<endl;
    cout<< " Enter a line or press "  << ENDFILE << ": ";
    cin.get(next);                      //Get next character.
} // end outer while
cout<< endl <<endl<<"Number of lines processed is "
    << lineCount <<endl;
return 0;
}
```

```
Enter a line or press CTRL-Z : This is one.
This is one.
```

```
Number of characters is 12
Enter a line or press CTRL-Z: Hello!
Hello!
Number of characters is 6
Enter a line or press CTRL-Z: CTRL-Z
Number of lines processed is 2
```

8.2 External Files

การอ่านข้อมูลและการแสดงผลข้อมูลในหัวข้อที่แล้ว จะเหมาะสำหรับโปรแกรมขนาดเล็ก ในกรณีของโปรแกรมที่มีขนาดใหญ่เรานิยมเก็บในแฟ้มภายนอก ในหน่วยความจำสำรอง เช่น ดิสก์ เราสามารถจะสร้างแฟ้มข้อมูลโดยใช้ โปรแกรม EDITOR ในการเข้าถึงแฟ้มข้อมูลเราต้องทราบชื่อสารบัญของแฟ้มเพื่อที่จะกำหนดเส้นทางในการเข้าถึงแฟ้มข้อมูลได้อย่างถูกต้อง การอ่านและการบันทึกข้อมูลจากแฟ้มข้อมูลภายนอกนี้เราต้องทำการการเชื่อมโยง Stream Object ไปยังไฟล์นั้น ภาษา C++ เราใช้คำสั่ง Open ในการเชื่อมโยง Stream Object ไปยังแฟ้มข้อมูลภายนอก โดยใช้ฟังก์ชั่นในการจัดการแฟ้มข้อมูล ดังนี้

Member Function	Purpose
fs.open(fname);	เป็นการเปิดแฟ้มข้อมูลสำหรับอ่านหรือแสดงผลโดย Stream fs จะเชื่อมโยงกับแฟ้มข้อมูลภายนอกที่ชื่อ fname
fs.get(ch);	เป็นการดึงอักขระ 1 ตัว จาก Stream fs แล้วนำไปเก็บในตัวแปร ch
fs.put(ch);	เป็นการบันทึกตัวอักขระ ch ลงใน Stream fs
fs.close();	เป็นการยกเลิกการเชื่อมโยงระหว่าง Stream fs และแฟ้มข้อมูลภายนอก
fs.eof();	เป็นฟังก์ชั่นในการทดสอบว่า Stream fs นั้นยังมีข้อมูลอยู่อีกหรือไม่ กรณีถ้าไม่มีข้อมูลโปรแกรมจะอ่านอักขระ <eof> ซึ่ง

	เป็นสิ่งบ่งบอกว่าจบแฟ้มข้อมูลแล้ว การทำงานของฟังก์ชั่นนี้มีค่าเป็น True
fs.fail();	เป็นฟังก์ชั่นที่ทำการส่งค่า True กลับ ถ้าไม่สามารถทำการเปิดแฟ้มข้อมูลเพื่อใช้ในการประมวลผลได้

ก่อนที่จะอ่านหรือบันทึกแฟ้มข้อมูลภายนอก ต้องมีการประกาศ Stream Object ของแฟ้มข้อมูลดังนี้

```
ifstream ins;
ofstream outs;
```

โดย ifstream ก็หมายถึง input file Stream และ ofstream หมายถึง Output file Stream เป็นชนิดของข้อมูลที่ถูกกำหนดใน C++ ไลบารี่ ที่ชื่อ fstream เราสามารถใช้ฟังก์ชั่นในการทำงานกับ Object ได้ดังตัวอย่างต่อไปนี้

```
ins.open (inFile);
```

ฟังก์ชั่นนี้จะทำหน้าที่ติดต่อระหว่าง Stream ins กับแฟ้มข้อมูลภายนอกในที่นี้ก็คือ insFile เพื่อทำการประมวลผลในโปรแกรม ในการใช้ฟังก์ชั่นต่าง ๆ เหล่านี้ เราต้องมีการกำหนด #include <fstream> เพื่อให้ Complier ได้ทราบ

ตัวอย่างที่ 8.3 Copying a file

โปรแกรมต่อไปนี้เป็นการอ่านข้อมูลจากแฟ้มที่ชื่อ InData.txt นำไปบันทึกในแฟ้ม OutData.txt โดยผลลัพธ์จากการทำงาน ข้อมูลในแฟ้มทั้งสอง จะมีข้อมูลที่เหมือนกันทุกประการ

```
//File: copyFile.cpp
//Copies file inData.txt to file OutData.txt
#include <cstdlib>          //for the definition of EXIT_FAILURE
#include <fstream>          //for external file streams
using namespace std;


//Associate stream objects with external file names
```

```
#define inFile "InData.txt"        //directory name for inFile
#define outFile "OutData.txt"   //diretory name for outFile

//Functions used.....
//Copies one line of text
int copyLine (ifstream&, ofstream&);
int main()
{
        int lineCount;          //output: nember of lines processed
        ifstream ins;           //ins is an input stream
        ofstream outs;          //outs is an output stream

//Open input and output file, exit on any error.
ins.open (inFile);              //connects ins to file inFile
if (ins.fail())
{
        cerr<< "*** ERROR: Cannot open "<< inFile
            << " for input."<<endl;
        return EXIT_FAILURE;        //failure return
}
outs.open(outFile);     //connect outs to file outFile
if (outs.fail())
{
        cerr << "***** ERROR: Cannot open " << outFile
            << " for output."<< endl;
        return EXIT_FAILURE;    //failure return
}
```

```
//Copy each character from inData to outData.
lineCount =0;
while (! ins.eof())
{
        if (copyLine (ins, outs)  ! = 0)
                lineCount++;
}

//Display a message on the screen.
cout<<"Input file copied to output file." << endl;
cout<<lineCount<<"lines copied."<<endl;

ins.close();                    //close input file stream
outs.close();                   //close output file stream

return 0;                       //successful return
}

//Copy one line of text from one file to another
//Pre:          ins is opened for input and outs for output.
//Post:         Next line of ins is written to outs.
//              The last character processed from ins is <nwln>;
//              the last character written to outs is <nwln>.
//Returns:      The number of characters copied.
int  copyLine
        (ifstream&  ins,        //IN: ins stream
        ofstream&  outs)        //OUT: outs stream
```

```
{
        //Local data ....
        const char NWLN = '\n';         //newline character

        char nextCh;                    //inout: character buffer
        int charCount = 0;              //number of characters copied

        //Copy all data characters from stream ins to
        //        stream outs.
        ins.get(nextCh);
        while ( (nextCh != NWLN)  &&  !ins.eof())
        {
                outs.put (nextCh);
                charCount++;
                ins.get (nextCh);
        }       //end while
        // If last character read was NWLN write it to outs.
        if  (!ins.eof())
        {
                outs.put(NWLN);
                charCount++;
        }
        return charCount;
} //end copyLine
```

การทำงานของโปรแกรมนี้มีการกำหนด Object ที่ชื่อ ins. เพื่อทำหน้าที่ในการอ่านข้อมูลโดยเชื่อมกับแฟ้มข้อมูลภายนอกที่ชื่อ InData.txt และ Object ชื่อ outs. ทำหน้าที่ในการ

แสดงผลข้อมูล โดยเชื่อมต่อกับแฟ้มข้อมูลภายนอกที่ชื่อ OutData.txt ในขั้นตอนแรก จะทำการเปิดแฟ้มข้อมูลทั้งสองแฟ้ม โดยมีการตรวจสอบการเปิดแฟ้มโดยใช้ฟังก์ชั่น fail ถ้าไม่สามารถทำการเชื่อมโยงได้ โปรแกรมจะแสดงข้อความบ่งบอกถึงความผิดพลาดที่เกิดขึ้น และจะมีการส่งค่าความผิดพลาดกลับไปยังระบบ(EXIT_FAILURE) ในกรณีที่มีการเชื่อมโยงแฟ้มข้อมูลได้สำเร็จ โปรแกรมจะมีการเรียกใช้ฟังก์ชั่นที่ชื่อ copyLine เพื่อทำการคัดลอกข้อมูลจากแฟ้ม InData.txt ครั้งละ 1 บรรทัด ไปยังแฟ้ม outData.txt จนกระทั่งหมดข้อมูลโดยการแสดงผลทางจอภาพจะปรากฏข้อความดังนี้

```
Input file copied to output file.
37 lines copied.
```

โดยค่า 37 เป็นจำนวนของบรรทัด ที่ทำการคัดลอกข้อมูลทั้งหมดจากแฟ้ม InData.txt ไปยัง OutData.txt สำหรับฟังก์ชั่น copyLine นี้เราสามารถใช้ฟังก์ชั่น getLine ในการอ่านข้อมูลจาก File stream ไปยัง Stream Object ได้ครั้งละ 1 บรรทัด เราสามารถที่จะแทนการทำงานวนรอบในฟังก์ชั่น main โดยเขียนได้ใหม่ดังนี้

```
string line;                //Next data line
lineCount = 0;
getline (ins, line);
while  (line.length() ! = 0)
{
        lineCount++;
        outs << line <<endl;
        getline (ins, line);
}
```

การทำงานจะทำการอ่านข้อมูล 1 บรรทัด เก็บใน Object string ที่ชื่อว่า line และมีการเพิ่ม lineCount ขึ้นหนึ่ง และทำการบันทึกข้อมูลนี้ใน Stream outs โดยทำงานวนรอบเพื่ออ่านข้อมูลจนกระทั่งขนาดของข้อมูลมีค่า = 0

กรณีศึกษา : แฟ้มข้อมูลเงินเดือน

โปรแกรมต่อไปนี้ เป็นโปรแกรมที่แสดงถึงการติดต่อสื่อสารระหว่างแฟ้มข้อมูลที่เก็บในดิสก์ กับโปรแกรมประยุกต์

ปัญหา ระบบเงินเดือนของบริษัทแห่งหนึ่งต้องการประมวลผลแฟ้มข้อมูล 2 แฟ้ม โดยแฟ้มแรกจะเก็บข้อมูลเงินเดือนพนักงาน ประกอบด้วย ชื่อ-นามสกุล ของพนักงาน, จำนวนชั่วโมงทำงาน และอัตราค่าจ้างต่อชั่วโมง สำหรับแฟ้มที่สองจะเกิดจากการประมวลข้อมูลในแฟ้มแรกโดยอ่านข้อมูลของพนักงานแต่ละคน เพื่อนำมาคำนวณเงินเดือน โดยแฟ้มข้อมูลที่สองประกอบด้วย ชื่อ-นามสกุลของพนักงาน และเงินเดือนที่ได้รับของพนักงาน

วิเคราะห์ การปัญหาเราสามารถวิเคราะห์ในเรื่องของความต้องการได้ดังนี้

DATA REQUIREMENTS

Streams Used

ifstream eds *//employee data information*

ofstream pds *//payroll data information*

Problem Input (from stream eds)

for each employee:

string firstName

string lastname

float hoursWorked

double hourlyRate

Problem Output(to stream pds)

for each employee:

string firstName

string lastname

double salary

Problem Output(to stream cout)

double totalPayroll *// total company payroll*

ออกแบบ จากการวิเคราะห์ถึงปัญหาต่างๆ สามารถออกแบบโดยเตรียม Stream ต่างๆ ที่เกี่ยวข้องกับแฟ้ม ในการอ่านและแสดงผลข้อมูลในฟังก์ชั่น main นอกจากนี้มีการสร้างฟังก์ชั่น processEmp เพื่อประมวลผลข้อมูลเพื่อหาเงินเดือนของพนักงานในบริษัททั้งหมด

ALGORITHM

1. เตรียม Stream ที่เกี่ยวข้องกับแฟ้มข้อมูลทั้งหมด
2. ประมวลผลข้อมูลพนักงานทั้งหมดและทำการหาผลรวมของเงินเดือน
3. แสดงผลรวมของเงินเดือนทั้งหมด

การวิเคราะห์และออกแบบฟังก์ชั่น processEmp

ฟังก์ชั่นนี้จะทำการอ่านระเบียนข้อมูลของพนักงานแต่ละคนจาก Input File ทำการประมวลผลเพื่อบันทึกระเบียนผลลัพธ์ไปยัง Output File โดยมีอัลกอริทึมดังต่อไปนี้

INTERFACE FOR processEmp

Input Arguments

Ifstream eds *//input stream-employee data stream*

ofstream pds *//output stream-payroll data stream*

Output Arguments

(none)

Function Return Value

double totalPayroll *//total company payroll*

การออกแบบโปรแกรม

```
//File: payrollFile.cpp
//Creates a company employee payroll file
//        computes total company payroll amount

#include <fstream>              //required for file streams
#include <cstdlib>              //for definition of EXIT_FAILURE
using namespace std;

//Associate streams with external file names
#define inFile "EmpFile.txt"    //employee file
#define outFile "Salary.txt"    //payroll file

//Functions used .....
//PROCESS ALL EMPLOYEES ND COMPUTE TOTAL
double processEmp (ifstream&, ofstream&);

int main()
{
        ifstream eds;           // input: employee data stream
        ofstream pds;           // output: payroll data stream
        double totalPayroll;    // output: total payroll

// Prepare files.
eds.open (inFile);
if  (eds.fail () )
{
```

```
        cerr <<  "**** ERROR: Cannot open " << inFile
             << " for input." << endl;
        return EXIT_FAILURE;  //failure return
}
pds.open (outFile);
if  (pds.fail () )
{
        cerr << "***ERROR: Cannot open " << outFile
             << " for output." << endl;
        eds.close();
        return EXIT_FAILURE;          // failure return
}
// Process all employees and compute total payroll.
totalPayroll = processEmp (eds, pds) ;

// Display result.
cout << "Total payroll is " << totalPayroll  << endl;

// Close files.
eds.close () ;
pds.close () ;
return 0;
}

//Process all employees and compute total payroll amount
//Pre:  eds and pds are prepared for input/output.
//Post: Employee names and salaries are written from eds to pds
```

```
//      and the sum of their salaries is returned.
//Returns: Total company payroll

double processEmp
        (ifstream& eds,             // IN: employee file stream
        ofstream& pds)              // IN: payroll file stream
{
        string  firstName;          // input: employee first name
        string lastName;            // input: employee last name
        float hours;                // input: hours worked
        double rate;                // input: hourly rate
        double salary;              // output: gross salary
        double payroll;             // return value – total company payroll
        payroll = 0.0;
        // Read first employee's data record.
        eds >> firstName >> lastName >> hours >>rate ;
        while ( ! eds.eof () )
        {
                salary = hours * rate;
                pds << firstName << " " <<lastName
                    << "  " << salary << endl;
                payroll += salary;
                //Read next employee 's data record.
                eds >> firstName >> lastName >> hours >>rate;
        } // end while
        return payroll;
} // processEmp
```

จากตัวอย่างถ้าข้อมูลในแฟ้ม EmpFile.txt ประกอบด้วยดังนี้

Jim Baxter 35.5 7.25 <nwln>

Adrian Cybriwsky 40.0 6.50<nwln>

Ayisha Mertens 20.0 8.00<nwln><eof>

เมื่อปฏิบัติงานตามโปรแกรมส่งผลให้แฟ้ม Salary.txt ประกอบด้วยข้อมูลดังนี้

Jim Baxter 257.38<nwln>

Adrian Cybriwsky 260.00<nwln>

Ayisha Mertens 160.00<nwln><eof>

และจะแสดงผลลัพธ์ออกทางจอภาพดังนี้

Total payroll is 677.38

แต่ถ้าข้อมูลในแฟ้ม EmpFile.txt ประกอบด้วยดังนี้

Jim Baxter# 35.5 7.25 <nwln>

Adrian Cybriwsky# 40.0 6.50<nwln>

Ayisha Mertens# 20.0 8.00<nwln><eof>

การอ่านโปรแกรมจากเดิมเราจะใช้คำสั่ง

```
eds >> firstName >> lastName >>hours >>rate;
```

สามารถเปลี่ยนเป็นการใช้คำสั่ง getline แทนได้ดังนี้

```
getline (eds, name, '#');
```

การทำงานของคำสั่งจะทำการดึงอักขระจาก input stream eds เป็นลำดับจนกระทั่งพบเครื่องหมาย # และจะนำอักขระทั้งหมดยกเว้นอักขระ # ให้แก่ string object name แต่ถ้าข้อมูลของแฟ้ม EmpFile.txt ประกอบด้วยดังนี้

```
Jim Baxter <nwln>
35.5   7.25 <nwln>
Adrian Cybriwsky<nwln>
40.0   6.50<nwln>
Ayisha Mertens<nwln>
20.0    8.00<nwln><eof>
```

เราสามารถประมวลผลทุก เรคอร์ด ในแฟ้ม โดยมีการปรับเปลี่ยนคำสั่งในการทำงานวนรอบดังนี้

```
payroll = 0.0;
// Get name of first employee.
getline (eds, name) ;
while  ( ! eds.eof () )
{
        // Get current employee salary data.
        eds  >> hours >> rate;
        salary = hours * rate;
        pds << name << " " << salary << endl;
        payroll += salary;

        // Get name of next employee.
        getline (eds, name) ;
}   // end while
```

การทำงานของอัลกอริทึมนี้อาจเกิดปัญหาในการอ่านข้อมูล เราสามารถหลีกเลี่ยงปัญหาที่อาจเกิดข้อผิดพลาดขึ้นโดยใช้ฟังก์ชั่น ignore ก่อนเรียกใช้คำสั่ง getline ในคำสั่ง while loop ดังนี้

```
// Get name of next employee.
eds.ignore (100, ' \n ');          //skip through first ' \n '
getline (eds, name) ;
```

กรณีศึกษา : การหาคะแนนเฉลี่ย

โปรแกรมต่อไปนี้เป็นโปรแกรมที่มีการอ่านข้อมูลจาก 2 แฟ้มข้อมูล ซึ่งเก็บชื่อวิชา และคะแนนสอบดังมีรูปแบบดังนี้

courseNo score1,score2,.....,scoreN, -999

โดยนำมาประมวลผลเพื่อหาค่าเฉลี่ยของแต่ละรายวิชา เก็บในแฟ้มผลลัพธ์ที่มีข้อมูลดังนี้

Course No	Group No	Course Average
CSC	1	83.71
	2	80.82
ENG	1	82.00
	2	78.20

Avg for group 1: 82.04

Avg for group 2: 82.01

นอกจากนี้โปรแกรมยังนำคะแนนเฉลี่ยของแต่ละรายวิชามาสร้างเป็นกราฟแท่ง

```
// Program :  Comparison of class averages
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
using  namespace std;
```

```
// function prototypes
void calculateAverage (ifstream& inp, double& courseAvg) ;
void printResult (ofstream& outp, string courseId,int groupNo, double avg);
int main()
{
        string  courseId1;              //course ID for group 1
        string  courseId2;              //course ID for group 2
        int numberOfCourses;
        double avg1;                    //average for a course in group 1
        double avg2;                    //average for a course in group 2
        double avgGroup1;               //average group 1
        double avgGroup2;               //average group 2
        ifstream group1;                //input stream variable for group 1
        ifstream group2;                //input stream variable for group 2
        ofstream outfile;               //output stream variable
        group1.open ("a:\\group1.txt") ;
        group2.open ("a:\\group2.txt") ;
        if ( ! group1 ||  ! group2)
        {
                cout << "Unable to open the files." << endl;
                return 1;
        }
        outfile.open("a:\\student.out") ;
        outfile << fixed << showpoint ;
        outfile << setprecision (2) ;
        avgGroup1 = 0.0 ;
        avgGroup2 = 0.0 ;
```

```
numberOfCourses = 0 ;
        //print heading :
outfile << "Course No   Group No     Course Average" << endl;
group1 >> courseId1;
group2 >> courseId2;
while (group1  &&  group2)
{
        if (courseId1 != courseId2)
        {
            cout <<"Data error: Course IDs do not match." << endl;
            cout << "Program terminates." << endl;
            return 1;
        }
        else
        {
            calculateAverage (group1, avg1) ;
            calculateAverage (group2, avg2) ;
            printResult (outfile, courseId1,  1,  avg1) ;
            printResult (outfile, courseId2,  2,  avg2) ;
            avgGroup1 = avgGroup1 + avg1 ;
            avgGroup2 = avgGroup2 + avg2 ;
            outfile <<endl;
            numberOfCourses++ ;
        }
        group1 >> courseId1 ;
        group2 >> courseId2 ;
}
```

```
        if (group1  &&  ! group2)
        {
                cout << "Ran out of data for group 2 before group1."
                        << endl;
        else
            if ( ! group1  &&  group2)
                cout<<"Ran out of data for group 1 before group 2."
            else
            {
                outfile << "Avg for group 1:"
                        << avgGroup1 / numberOfCourses << endl ;
                outfile << "Avg for group 2: "
                         << avgGroup2 / numberOfCourses << endl;
            }
        group1.close() ;
        group2.close() ;
        outfile.close() ;
        return 0;
}

void calculateAverage (ifstream& inp, double& courseAvg)
{
        double totalScore = 0.0 ;
        int numberOfStudents = 0 ;
        int score;

        inp >> score ;
```

```
        while (score !=  -999)
        {
                totalScore = totalScore + score ;
                numberOfStudents++;
                inp >> score ;
        } // end while
        courseAvg = totalScore / numberOfStudents ;
} // end calculate Average

void  printResult (ofstream& outp, string courseID, int groupNo, double avg)
{
        if (groupNo ==1 )
                outp << "  " << courseID << "     " ;
        else
                outp << "                 ";
        outp << setw (8) << groupNo << setw(17) << avg << endl;
} // end printResult
```

โปรแกรมนี้มีการทำงานตามขั้นตอนดังนี้

1. กำหนดตัวแปรเริ่มต้นโดย group1 และ group2 เป็น input stream ตัวแปร outfile เป็นตัวแปร output stream

2. ทำการเปิดแฟ้ม group1.txt และ group2.txt เพื่ออ่านรหัสรายวิชาในกรณีที่ไม่สามารถเปิดแฟ้มได้หรือรหัสรายวิชามีค่าแตกต่างกันจะมีข้อความบ่งบอกถึงข้อผิดพลาดที่เกิดขึ้นและส่งค่า 1 กลับไปยังฟังก์ชั่น main

3. คำนวณหาค่าเฉลี่ยของแต่ละรายวิชาของ group1 และ group2 โดยการเรียกใช้ฟังก์ชั่น calculateAverage

4. บันทึกผลลัพธ์จากการทำงานลงบนแฟ้ม student.out

5. กระทำซ้ำข้อ 3-5 จนกระทั่งหมดข้อมูล

6. บันทึกข้อมูลค่าเฉลี่ยของแต่ group

ในกรณีที่ต้องการบันทึกข้อมูลเป็นกราฟแท่ง ซึ่งแสดงถึงค่าเฉลี่ยของแต่ละกลุ่มของแต่ละรายวิชาลงแฟ้ม student.out เราสามารถปรับเปลี่ยนฟังก์ชั่น printresult ได้ดังนี้

```
void printResult (ofstream& outp, string courseID, int groupNo, double avg)
{
        int noOfSymbols ;
        int count;

        if (groupNo == 1)
            outp << setw (4) << courseID << "     ";
        else
            outp << "              ";

        noOfSymbols = static_cast<int>(avg) / 2 ;
        if  (groupNo == 1)
                for (count = 1; count <= noOfSymbols; count++)
                    outp << '*' ;
        else
                for (count =1 ; count <= noOfSymbols; count++)
                    outp << '#' ;

        outp << endl;
} // end printResults
```

เราสามารถกำหนดฟังก์ชั่นชื่อ printHeading ในการพิมพ์หัวเรื่องของกราฟ ได้ดังนี้

```
void printHeading (ofstream& outp)
{
        outp << "Course          Course Average"  << endl ;
        outp << "  ID  0       10      20      30      40      50      60"
             << "  70   80      90      100" <<endl ;
        outp << "      |.....|.....|.....|.....|.....|.....|.....| "
             << ".....|.....|.....|.....|.....| " << endl ;
} // end printHeading
```

สรุป

1. Streams เป็นลำดับของข้อมูลที่นำเข้าหรือแสดงผลในโปรแกรม มาตรฐานของ input Stream มีชื่อว่า cin และมาตรฐานของ output Stream มีชื่อว่า cout ซึ่งถูกนำมาเก็บรวมกันใน Library ที่ชื่อว่า iostream
2. cin จะเป็นการเชื่อมต่อกับแป้นพิมพ์ เมื่อโปรแกรมนั้นมีการอ่านอักขระ
3. cout จะเป็นการเชื่อมต่อกับจอแสดงผล เมื่อมีการพิมพ์ตัวอักษรออกทางจอแสดงผล
4. cin จะมีเบื้องหลังการทำงานคือในระหว่างที่ป้อนข้อมูลตัวกระทำที่เรียกว่า Stream Extraction operator (>>) จะทำการแปลงลำดับของข้อมูลให้อยู่ในรูปแบบที่เหมาะสม
5. การแสดงผลเราใช้ตัวกระทำที่เรียกว่า Insertion operator (<<) ในการแปลงค่าที่เก็บภายในหน่วยความจำให้เป็นลำดับของตัวอักษรเพื่อแสดงออกทางจอแสดงผล
6. โปรแกรมที่มีขนาดใหญ่เรานิยมเก็บในแฟ้มภายนอก ในหน่วยความจำสำรอง เช่น ดิสก์ การอ่านและการบันทึกข้อมูลจากแฟ้มข้อมูลภายนอก ต้องทำการการเชื่อมโยง Stream Object ไปยังไฟล์นั้น ภาษา C++ เราใช้คำสั่ง Open
7. การอ่านหรือบันทึกแฟ้มข้อมูลภายนอก ต้องมีการประกาศ Stream Object ของแฟ้มข้อมูลดังนี้

```
ifstream ins;
ofstream outs;
```

   โดย ifstream ก็หมายถึง input file Stream และ ofstream หมายถึง Output file Stream เป็นชนิดของข้อมูลที่ถูกกำหนดใน C++ ไลบารี่ ที่ชื่อ fstream
8. เราสามารถใช้ฟังก์ชั่นในการทำงานกับ Object ได้ดังตัวอย่างต่อไปนี้

```
ins.open (inFile);
```

   ฟังก์ชั่นนี้จะทำหน้าติดต่อระหว่าง Stream ins กับแฟ้มข้อมูลภายนอกในที่นี้ก็คือ inFile เพื่อทำการประมวลผลในโปรแกรม
9. ในการใช้ฟังก์ชั่นต่าง ๆ เหล่านี้ เราต้องมีการกำหนด #include <fstream> เพื่อให้ Complier ได้ทราบ

แบบฝึกหัด

1. จงเขียนโปรแกรมอ่านข้อมูลหลายๆบรรทัดจากแฟ้มข้อมูลหนึ่ง โดยให้โปรแกรมสามารถนับจำนวนคำในแต่ละบรรทัด และพิมพ์จำนวนของคำทั้งหมดในแฟ้มนี้พิมพ์ออกมาทางจอภาพ
2. จงเขียนโปรแกรมคิดดอกเบี้ยจากระเบียนข้อมูลที่เก็บในแฟ้ม โดยข้อมูลที่เก็บในแฟ้มประกอบด้วยเงินต้น ,จำนวนเดือนที่กู้ ,และอัตราดอกเบี้ยต่อปี จงอ่านข้อมูลจากแฟ้มและนำมาคิดหาดอกเบี้ยที่ต้องจ่ายทั้งหมด
3. จงเขียนโปรแกรมอ่านกลุ่มของตัวอักษรที่เป็นจำนวนเงินของโรมัน โดยทำการแปลงให้เป็นรูปแบบของตัวเลขในรูปของ Arabic โดยความสัมพันธ์ของตัวเลขเป็นดังนี้

Roman	Arabic
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

ตัวอย่าง ถ้าผู้ใช้ป้อนข้อมูล LXXXVII ผลของการทำงานมีค่าเท่ากับ 87 เป็นต้น