บทที่ 12
การจัดการข้อมูลแบบไฟล์

❒ ความหมายของไฟล์

ไฟล์(File) หมายถึง โครงสร้างของกลุ่มข้อมูลที่จัดเก็บในรูปแบบที่คอมพิวเตอร์สามารถนำมาใช้งานได้ ไฟล์อาจถูกจัดเก็บอยู่ในหน่วยความจำหรือสื่อบันทึกข้อมูล เช่น แผ่นดิสก์ ฮาร์ดดิสก์ หรือ CD-ROM ซึ่งสามารถนำข้อมูลในไฟล์ที่บันทึกในสื่อดังกล่าวกลับมาใช้ใหม่ได้อีก นอกจากนั้นข้อมูลในไฟล์สามารถส่งออกไปแสดงผลที่จอภาพ หรือเครื่องพิมพ์ โดยอาศัยการจัดการด้วยโปรแกรมที่มีคำสั่งเกี่ยวกับการจัดการไฟล์โดยเฉพาะ

❒ ชนิดของไฟล์ใน C++

1. Text File หมายถึง ไฟล์ที่ประกอบด้วยตัวอักษร เครื่องหมายต่าง ๆ ที่ใช้ในข้อความ มีการกำหนดรหัสการจบแต่ละบรรทัดด้วยรหัสแป้น enter หรือรหัสการขึ้นบรรทัดใหม่(carriage return) และมีเครื่องหมายจบไฟล์ (end of file maker) โดยทั่วไป Text File จะเป็นไฟล์ที่สร้างขึ้นจากโปรแกรมประเภท Text Editor หรือ Word Processor เช่น ไฟล์ชนิด
.CPP ก็สร้างจาก Text Editor ของ C++ ดังนั้นจึงสามารถ Text File ไปเปิดในโปแกรมประเภท Word Processor ได้ ใช้คำสั่ง Type ของ DOS ดูรายละเอียดในไฟล์ได้

2. Binary File หมายถึง ไฟล์ที่ประกอบด้วยของมูลประเภทโครงสร้าง (Structure) หรือมีการบีบลดขนาด (Compress) ซึ่งทำให้ข้อมูลมีลักษณะแตกต่างจากข้อมูลใน Text File
การสร้าง Binary File จะสร้างจากโปรแกรมสำหรับไฟล์ชนิดนั้นโดยเฉพาะ เพื่อความสะดวกในการอ่านและเขียนไฟล์ เช่น ไฟล์ในโปรแกรม dBASE, Lotus เป็นต้น

❒ การดำเนินการเกี่ยวกับไฟล์

ใน C++ มีคลาสที่ทำหน้าที่เกี่ยวกับการรับข้อมูลและการนำข้อมูลออกจากไฟล์ (File I/O) อยู่ 3 คลาส ได้แก่

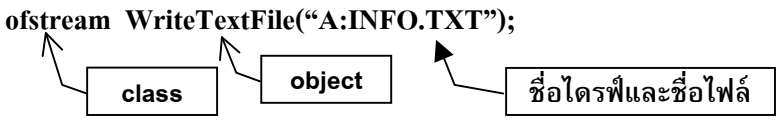
- คลาส ifstream ทำหน้าที่รับข้อมูลจากไฟล์เข้ามาในหน่วยความจำ
- คลาส ofstream ทำหน้าที่นำข้อมูลจากหน่วยความจำส่งออกไปที่ไฟล์
- คลาส fstream ทำหน้าที่ทั้งรับข้อมูลจากไฟล์เข้ามาในหน่วยความจำและนำข้อมูลจากหน่วยความจำออกไปที่ไฟล์

ทั้ง 3 คลาส คือ ifstream, ofstream, stream เก็บอยู่ในไฟล์ fstream.h ดังนั้นโปรแกรมที่มีการดำเนินการเกี่ยวกับข้อมูลประเภทไฟล์ จะต้องมีการ include เฮดเตอร์ไฟล์ ที่ชื่อ **fstream.h** ด้วยเสมอ

ในการดำเนินการเกี่ยวกับไฟล์ ทั้ง Text File และ Binary File มีกิจกรรมหลักๆ ที่ต้องดำเนินการ คือ ***การบันทึกข้อมูลจากหน่วยความจำเข้าเก็บในไฟล์ และการอ่านข้อมูลจากไฟล์มาไว้ในหน่วยความจำ***

❒ วิธีดำเนินการกับไฟล์ข้อมูลสตริง(Text File)

1. การบันทึกสตริงเข้าไฟล์ในดิสก์ ทำได้โดยการสร้างออบเจ็กต์ให้เป็นสมาชิกของคลาส ofstream และกำหนดชื่อไฟล์ ตามรูปแบบตัวอย่าง



จากรูปแบบ ofstream คือชื่อคลาส WriteTextFile คือชื่อออบเจ็กต์ที่สร้างขึ้นเพื่อทำหน้าที่บันทึกไฟล์และในวงเล็บ ("test1.txt") คือชื่อไฟล์ ซึ่งเป็น Text File ที่จะใช้บันทึกข้อความ ถ้าไม่ระบุไดเรกทอรี่จะบันทึกลงในไดเรกทอรีปัจจุบัน เราสามารถเขียนออกเจ็กต์สำหรับทำหน้าที่บันทึกสตริงเข้าไฟล์ในลักษณะอื่นๆ ได้ เช่น

ofstream SaveTextFile("A:Test.txt");

ofstream write_text_file("A:\DATA\test.dat");

วิธีการบันทึกสตริงเข้าเก็บไว้ในไฟล์ เขียนได้ดังนี้

WriteTextFile<<"Hello, I love C++ and Computer\n";

หมายถึงให้บันทึกข้อความ Hello, I love C++ and Computer ไว้ในไฟล์ data.txt เป็นต้น

ตัวอย่างโปรแกรม TEXTFILE.CPP แสดงการบันทึกข้อความเข้าเก็บไว้ในไฟล์ ชื่อ INFO.TXT ในไดเรกทอรี A: การแสดงผลลัพธ์ของโปรแกรมจะทำการบันทึกข้อความที่กำหนดไว้ในไฟล์ INFO.TXT ซึ่งเราสามารถดูเนื้อหาในไฟล์ได้ โดยการใช้คำสั่ง ใน DOS prompt

A:\>TYPE INFO.TXT [กด enter]

หรือเปิดไฟล์โดยใช้ Editor ของ C++ โดยใช้คำสั่ง File / Open พิมพ์ชื่อไฟล์ A:INFO.TXT แล้ว OK หรือ enter

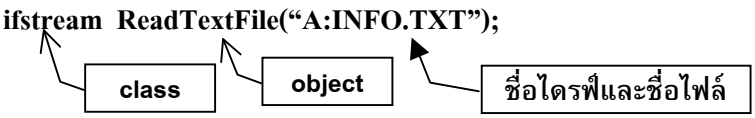
```
/*Program: TextFile.CPP
  Process: write text line to text file  A:INFO.TXT
*/
#include <iostream.h>
#include <fstream.h>

void main()
{ //create WriteTextFile object  from  ofstream class
  ofstream WriteTextFile("A:INFO.TXT");
  WriteTextFile<<" Hello, How are you today? \n";
  WriteTextFile<<" How do you feel about C++? \n";
  WriteTextFile<<" I hope that you will fall in love it.\n";
  WriteTextFile<<" Are you O.K. ";
}
```

ข้อสังเกต การใช้คลาส ofstream สร้างออบเจ็กต์ เพื่อบันทึกสตริงก์ ออปเจ็กต์จะทำการเปิดบันทึกข้อมูลแล้วปิดไฟล์ให้โดยอัตโนมัติ

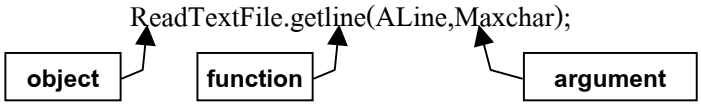
2. การอ่านสตริงจากไฟล์ในดิสก์ ทำได้โดยการ

2.1 ต้องสร้างออบเจ็กต์ให้เป็นสมาชิกของคลาส ifstream และกำหนดชื่อไฟล์ที่ต้องการอ่านข้อมูล ตามรูปแบบดังนี้



หมายถึง สร้างออบเจ็กต์ชื่อ **ReadTextFile** เป็นสมาชิกของคลาสifstream เพื่อดำเนินการอ่านสตริงจากไฟล์ชื่อชื่อ Test1.txt จากไดเรกทอรี่ปัจจุบัน

2.2 ใช้ฟังก์ชัน getline() ที่เป็นสมาชิกของคลาส istream ซึ่งเป็นดีไรฟ์คลาสของ ifstream ฟังก์ชัน getline() ทำหน้าที่อ่านอักษรจากไฟล์มาเก็บไว้ในตัวแปรประเภท charทีละบรรทัด ('\n') จำนวนสูงสุดตามที่กำหนด ตามรูปแบบดังนี้



หมายถึง ออบเจ็กต์ ReadTextFile มีการอ่านสตริงจากไฟล์ Test1.txt ด้วยฟังก์ชัน getline() อ่านมาทีละบรรทัดมาเก็บไว้ที่ตัวแปร ALine จำนวนที่เก็บได้สูงสุดในแต่ละบรรทัดเท่ากับ Maxchar

- ***ตัวอย่างโปแกรม READTXT1.CPP ต่อไปนี้ ทำหน้าที่ในการอ่านข้อมูลจาก Text File ชื่อ INFO.TXT ขึ้นมาทีละบรรทัด แต่ละบรรทัดไม่เกิน 80 ตัวอักษร จนกว่าจะจบไฟล์หรือจนกว่าจะอ่านข้อมูลไม่ได้อีกแล้ว***

```
/*Program: TXTREAD1.CPP
  Process: read text file to string of character or to line
*/
#include <fstream.h>
#include <iostream.h>
#include <conio.h>

void main()
{ char text[80];
  clrscr();
  //ReadTextFile() is object that user create from ofstream class
  ifstream ReadTextFile("A:INFO.TXT");
  while(ReadTextFile)
  {
    ReadTextFile.getline(text,80);
    cout<<text<<endl;
  }
  cout<<"this is text that read from A:INFO.TXT file\a";
  getch();
}
```

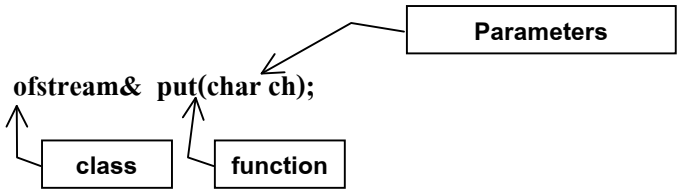
<u>หมายเหตุ</u>

while(ReadTextFile) เป็นคำสั่งแบบลูป(Looping) ทำหน้าที่อ่านข้อมูลจนจบไฟล์ โดยฟังก์ชัน getline() จะอ่านข้อมูล 1 บรรทัดมาเก็บไว้ที่ตัวแปร text แล้วแสดงออกไปที่จอภาพด้วยคำสั่ง cout<< ตามเงื่อนไข ตรวจสอบการจบไฟล์ด้วยออบเจ็กต์ ReadTextFile มีการทำงานดังนี้

- ถ้าจบไฟล์ ReadTextFile จะให้ค่าเป็น 0
- ถ้ายังไม่จบไฟล์ ReadTextFile จะให้ค่าที่ไม่ใช่ 0

❒ วิธีดำเนินการกับไฟล์อักษร

1. การบันทึกตัวอักษรเข้าไฟล์ในดิสก์ ทำได้โดยใช้ฟังช์ชัน put() ซึ่งเป็นฟังก์ชันในคลาส ostream ใช้สำหรับบันทึกอักษรเข้าไฟล์ครั้งละ 1 ตัวอักษร มีรูปแบบของฟังก์ชัน put()ดังนี้



ฟังก์ชัน put() จะทำหน้าที่ส่ง ch ไปบันทึกในไฟล์ที่กำหนด

ตัวอย่างโปรแกรม CharFil1.CPP ใช้ฟังก์ชัน put() ทำการบันทึกตัวอักษรจากข้อความครั้งละ 1 ตัวอักษรบันทึกลงในไฟล์ A:INFO2.TXT

```
/*Program: CharFil1.CPP
  Process: write character to  file
*/
#include <fstream.h>
#include <string.h>
void main()
{  char Str[]="I love to learn C++ Language";
   ofstream WriteTextFile("A:INFO2.TXT");
   for(int n=0;n<strlen(Str);n++)
      WriteTextFile.put(Str[n]);
}
```

- ***ตัวอย่างโปรแกรม WriteCha.CPP ทำหน้าที่รับข้อมูลครั้งละตัวอักษรและเขียนลงไฟล์A:INFO3 จนกว่าจะกดแป้น Enter จากนั้นอ่านข้อมูลจากไฟล์***

```
/*Program: WriteCha.CPP
  Process: write for each character to text file by put() function
*/
#include <fstream.h>
#include <iostream.h>
#include <conio.h>
#include <string.h>
void main()
{ char ch;
  clrscr();
  ofstream WriteTextFile("A:INFO3.TXT");
  cout<<"**** Please key character ....press enter to stop ****";
  cout<<endl;
while(ch!=13)
  { ch=getche();  //input 1 character to ch
    WriteTextFile.put(ch); //write to file
  }
  WriteTextFile.close();
  clrscr();
  //ReadTextLine() is object that user create from ifstream
  char chr;
  ifstream ReadTextLine("A:INFO3.TXT");
  while(ReadTextLine);
{
     ReadTextLine.get(chr);
     cout<<chr;
   }
  cout<<endl<<endl<<"....this is text that read from INFO3.TXT
  getch();
}
```

❒ วิธีดำเนินการกับไบนารีไฟล์ (Binary File)

Binary File เป็นไฟล์ที่ประกอบด้วยของมูลประเภทโครงสร้าง (Structure) หรือมีการบีบลดขนาด (Compress) ไฟล์ประเภทนี้จะใช้กับข้อมูลชนิด structure มีการจัดการ ดังนี้

1. การเขียนข้อมูลลงไฟล์ในดิสก์ ใช้ฟังก์ชัน write() ซึ่งเป็นฟังก์ชันในคลาส ostream ฟังก์ชัน write() มีรูปแบบดังนี้

ostream& write(const char* , int n);

ฟังก์ชัน write() จะนำตัวอักษรจากพารามิเตอร์ตัวที่ 1 จำนวน n ตัวอักษาไปเขียนไฟล์ในไฟล์ โดยในจำนวน n ตัวนี้รวมอักษร Null ไว้แล้ว

2. การอ่านข้อมูลจากไฟล์ ทำได้โดยใช้ฟังก์ชัน read() ซึ่งเป็นฟังก์ชันในคลาส istream มีรูปแบบดังนี้

istream& read(const char* , int n);

ฟังก์ชัน read() จะอ่านอักษรจากไฟล์มาเก็บไว้ที่พารามิเตอร์ตัวที่ 1 โดยมีจำนวนอักษรตามที่กำหนดไว้ในพารามิเตอร์ตัวที่ 2

- ***ตัวอย่างโปรแกรม STD_FILE.CPP มีการสร้าง class และข้อมูลชนิด structure เพื่อนำไปบันทึกในไฟล์ข้อมูลในไดรฟ์ A: ชื่อ student.dat โดยบันทึกข้อมูลได้เพียงครั้งละ 1 ชุดหรือ 1 เรคอร์ด (record)***

```
/*Program: STD_FILE.CPP
 Process:Create Class
         Create Binary File
         Read Binary File
         Create Menu() Function
*/
#include <fstream.h>
#include <iostream.h>
#include <conio.h>
#include <string.h>

//prototype function
void Menu();
void Create();
```

```
void Display();
//creat class
class STD
{
  protected:       // data member
     char code[9];
     char name[40];
     int  age;

  public:
      void InputData()   //member function
     {
           cout<<"Enter Code : ";cin>>code;
           cout<<"Enter Name : ";cin>>name;
           cout<<"Enter Age  : ";cin>>age;
     }

      void ShowData()   //memdber function
     {
           cout<<"Enter Code : "<<code<<endl;
           cout<<"Enter Name : "<<name<<endl;
           cout<<"Enter Age  : "<<age<<endl;
           getch();
     }
};

STD Student;  //create Object student from STD class
//begin main function
void main()
{
   Menu();
}

void Menu()   //user defined function
{  char choice=' ';
   while(choice!='3')
```

```
  { clrscr();
    gotoxy(25,2);cout<<"Main Menu  Student Data";
    gotoxy(30,4);cout<<" 1. Create File and Input Data";
    gotoxy(30,5);cout<<" 2. Displa Data File ";
    gotoxy(30,6);cout<<" 3. Exit ";
    gotoxy(30,8);cout<<"Select choice :  ";choice=getch();
    switch(choice)
    { case '1':
              Create();
              break;
      case '2':
              Display();
              break;
     case '3':
             break;
    }
  }
}

void  Create()//user defined function
{
  clrscr();
  Student.InputData();
  ofstream WriteFile("A:student.dat");     //create file
  WriteFile.write((char*)&Student,sizeof(Student));    //write data to file
}

void Display()//user defined function
{
  clrscr();
  ifstream ReadFile("A:student.dat");    //open file
  ReadFile.read((char*)&Student,sizeof(Student));   //read  data from file
  Student.ShowData();
  ReadFile.close();
}
```

❒ วิธีดำเนินการกับไฟล์พอยน์เตอร์ (File Pointer)

เนื่องจากการจัดการกับข้อมูลของไฟล์ประเภท Binary หลาย ๆ ออบเจ็กต์หรือหลาย ๆ เรคอร์ด ไม่สามารถทำได้หากใช้เฉพาะฟังก์ชัน write() หรือ read() เพียงอย่างเดียว แต่จะต้องใช้ File Pointer ในการจัดการ โดยใช้ฟังก์ชัน open() เปิดไฟล์ในโหมดต่าง ๆ ซึ่งมีรูปแบบดังนี้

void open(const char* name, int mode, int prot =filebuff::openprot);

โดยที่

name คือ ชื่อไฟล์ที่ต้องการเปิด

mode คือ โหมดการเปิดเพื่อวัตถุประสงค์ต่าง ๆ ดังตารางดังนี้

โหมด	ความหมาย
in	เปิดไฟล์เพื่ออ่านข้อมูล
out	เปิดไฟล์เพื่อบันทึกข้อมูล
app	เริ่มอ่านหรือบันทึกไฟล์ตอนท้ายไฟล์
ate	ลบไฟล์ก่อนที่จะอ่านหรือเซฟไฟล์
nocreate	จะเกิดความผิดพลาถ้าเปิดไฟล์ที่ไม่มีอยู่
noreplace	จะเกิดความผิดพลาดถ้าเปิดไฟล์ที่มีอยู่เพื่อเซฟข้อมูล
binary	เปิดไฟล์ไบนารี

โดยที่ โหมดเหล่านี้สามารถใช้ร่วมกันได้โดยใช้เครื่องหมาย | คั่นระหว่างแต่ละโหมด เช่น **open(“TEST.DAT”,ios::in|ios::out);**

prot คือ วิธีการติดต่อกับ DOS ซึ่งตามปกติถ้าไม่กำหนด ให้ติดต่อกับ DOS เพื่อบันทึกหรืออ่านข้อมูลจากไฟล์

<u>การสร้างไฟล์และการอ่านไฟล์พอยน์เตอร์ มีขั้นตอนสำคัญ ดังนี้</u>

1. สร้างคลาสหนดข้อมูลและฟังก์ชัน สร้างคลาสเพื่อกำหนดข้อมูลชนิด structure ภายในคลาส

พร้อมทั้งสร้าง Member Function ต่าง ๆ ที่ต้องการในโปรแกรม เช่น

```
class Person
  {
    protect:
        char  Name[50]
        int   Age;
    public:
        void  GetData()  //ฟังก์ชันทำหน้าที่รับข้อมูล
          {  cout<< "Enter  Name : "; cin>>Name;
             cout<< "Enter  Age : "; cin>>Age;
          }

        void  ShowData()   //ฟังก์ชันทำหน้าที่แสดงข้อมูล
          {  cout<<"Name : "<<Name<<endl;
             cout<<"Name : "<<Age<<endl;
          }
  };
```

2. สร้างออบเจ็กต์ที่เป็น record ข้อมูลจากคลาสที่สร้างไว้ในข้อ 1

```
Person  Student;        //Person คือ คลาส และ Student คือออบเจ็กต์
```

3. การสร้างออบเจ็กต์ประเภทไฟล์ คือ การสร้างออบเจ็กต์ประเภทไฟล์จากคลาส fstream

```
fstream  File;          //fsteram คือ คลาส และ File คือออบเจ็กต์ที่กำหนดขึ้น
```

2. การเปิดไฟล์ในโหมดต่าง ๆ ด้วยฟังก์ชัน open() โดยกำหนดโหมดตามความต้องการ ดังตัวอย่าง

```
File.open("A:TEST.DAT",ios::app|ios::out|ios::in);
```

3. การเขียนหรือบันทึกข้อมูลลงไฟล์ ด้วยฟังก์ชัน write() ดังนี้

```
File.write((char*)&Student,sizeof(Student));
```

4. การอ่านข้อมูลจากไฟล์ ด้วยฟังก์ชัน read() ดังนี้

```
File.read((char*)&Student,sizeof(Student));
```

5. การกำหนดตำแหน่งของข้อมูลในไฟล์ ด้วยฟังก์ชัน tellg(), seekg(), tellp(), seekp()

การจัดการข้อมูลในไฟล์ ณ ตำแหน่งต่าง ๆ ที่กำหนดของไฟล์ เรียกว่า ไฟล์พอยน์เตอร์ ต้องใช้ฟังก์ชันต่าง ๆ ที่เกี่ยวข้องกับการระบุตำแหน่งข้อมูลในไฟล์ มีรูปแบบ ดังนี้

long tellg(); ให้ค่าตำแหน่งของไฟล์พอยน์เตอร์ซึ่งเป็นตำแหน่งอ่านข้อมูล

istream& seekg(sreampos pos) ให้เลื่อนไฟล์พอยน์เตอร์ไปยังตำแหน่งที่ต้องการอ่านข้อมูล เช่น File.seekg(0) หมายถึงเลื่อนไปยังตำแหน่งเรคอร์ดหมายเลข 0 ซึ่งเป็นตำแหน่งแรกของไฟล์ ซึ่งเริ่มจาก 0, 1, 2, ...

streampos tellp(); ให้ค่าตำแหน่งของไฟล์พอยน์เตอร์ซึ่งเป็นตำแหน่งบันทึกข้อมูล

ostream& seekp(streampos); จะเลื่อนไปที่ตำแหน่งของไฟล์พอยน์เตอร์ซึ่งเป็นตำแหน่งบันทึกข้อมูล

ตัวอย่างโปรแกรม File_poin แสดงการใช้ File pointer จัดเก็บข้อมูลพร้อมกันได้หลายเรคอร์ด (Record) บันทึกข้อมูล, อ่านข้อมูลและค้นหาข้อมูลจากไฟล์

```
/*Program: FIL_POIN.CPP
  Process:Create Class
          Create Binary File by pointer
          Create Menu() Function
          Write,Read, Search Record
 */
#include <fstream.h>
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
//prototype function
void Menu();
void InputData();
void ReportData();
void SearchByCode();
//creat class
class STD
{
  public:       // data member
    char code[9];
    char name[40];
    int  age;

  public:
    void InputData()   //member function
```

```
    {
          cout<<"\nEnter Code : ";cin>>code;
          cout<<"Enter Name : ";cin>>name;
          cout<<"Enter Age  : ";cin>>age;
    }

     void ShowData()   //memdber function
    {
          cout<<"Code : "<<code<<endl;
          cout<<"Name : "<<name<<endl;
          cout<<"Age  : "<<age<<endl;

    }
};

//global variable
STD Student;  //create Object student from STD class
fstream File;  //creat Object File from fstream class

//begin main function
void main()
{
  Menu();
}

void Menu()
{  char choice=' ';
   while(choice!='4')
   { clrscr();
     gotoxy(25,2);cout<<"Main Menu Student Data <<A:Student.dat>>";
     gotoxy(30,4);cout<<"1. Appen Data ";
     gotoxy(30,5);cout<<"2. Report Data ";
     gotoxy(30,6);cout<<"3. Search Data from File ";
   //  gotoxy(30,7);cout<<"4. Search Data from File ";
     gotoxy(30,7);cout<<"4. Exit ";
     gotoxy(30,10);cout<<"Select choice :  ";choice=getch();
```

```
    switch(choice)
    {
      case '1':
            InputData();
            break;
      case '2':
            ReportData();
            break;
      case '3':
            SearchByCode();
      case '4':
            break;
    }
  }
}

void InputData()
{
  char ans=' ';
  File.open("A:student.dat",ios::app|ios::out|ios::in);
  clrscr();
  cout<<" Enter information of student: "<<endl;
  while(toupper(ans)!='N')
  {
    Student.InputData();
    File.write((char*)&Student,sizeof(Student));
    cout<<"Append data any more? <y/n> ? ";
    ans=getch();cout<<endl;
  }
  File.close();
}

void ReportData()
{  int all_rec=0,rec_no=1;
  //skip pointer of record at record #0
  clrscr();
```

```
  File.open("A:student.dat",ios::app|ios::out|ios::in);
  File.seekg(0,ios::end);
  all_rec=File.tellg()/sizeof(Student);
  File.seekg(0,ios::beg);
  cout<<"Report all record form file ...."<<endl;
  cout<<"Total  record : "<<all_rec<<endl;
  File.read((char*)&Student,sizeof(Student)); //read first record
  while(!File.eof())
  {  cout<<"record# "<<rec_no<<endl;
     Student.ShowData();
     File.read((char*)&Student,sizeof(Student));  //read next record
     cout<<"press any key..."<<endl<<endl;
     getch();rec_no++;
  }
  cout<<"end of file\a";getch();
  File.close();
}

void SearchByCode()
{ char co[9];
  int recno=0;
  cout<<"\n\nEnter code to search: ";cin>>co;
  File.open("A:student.dat",ios::in);
  File.seekg(0,ios::beg);
  File.read((char*)&Student,sizeof(Student)); //read record search no#
  while(!File.eof()&&strcmp(co,Student.code))
  {  //File.seekg((i+1)*sizeof(Student));
     File.read((char*)&Student,sizeof(Student)); //read record search no#
  }
  if((strcmp(co,Student.code)==0))
  {
     cout<<"\nRecord found #: "<<File.tellg()/sizeof(Student)<<endl;
     Student.ShowData();
  }
  else
     cout<<"Record not found!!!\a\a";
```

```
 cout<<"\npress any key to continue...";getch();
 File.close();
}
```

❒ แบบฝึกหัดท้ายบท

1. ให้เขียนโปรแกรมเพื่อเก็บประวัติพนักงานของบริษัทแห่งหนึ่ง รายละเอียดข้อมูลที่จัดเก็บได้แก่
   - รหัสพนักงาน
   - ชื่อสกุล
   - ที่อยู่
   - อายุ
   - เงินเดือน
   - ตำแหน่ง

   จากรายละเอียดของข้อมูลให้เขียนโปรแกรมโดยใช้ข้อมูลแบบไฟล์ ให้โปรแกรมมีฟังก์ชันต่าง ๆ ในโปรแกรมทำงานตามหน้าที่ ดังนี้
   - บันทึกประวัติพนักงาน
   - แสดงรายงานรายละเอียดประวัติของพนักงานทุกคนได้
   - แสดงรายงานเงินเดือนพนักงานแต่ละคนและ รายละเอียดการหักภาษี ณ ที่จ่าย 3% ของพนักงาน พร้อมยอดสรุปค่าเฉลี่ยเงินเดือนของพนักงาน
   - ค้นหาประวัติพนักงานเป็นรายบุคล โดยการกรอกรหัสพนักงานเพื่อค้นหาได้

2. ให้นักศึกษาเขียนโปรแกรมเพื่อคำนวณการตัดเกรดนักศึกษา โดยจัดเก็บข้อมูลเป็นชนิดไฟล์ข้อมูล กำหนดเงื่อนไขการตัดเกรด ดังนี้

คะแนนระหว่าง	1-49	เกรด	F
คะแนนระหว่าง	50-59	เกรด	D
คะแนนระหว่าง	60-69	เกรด	C
คะแนนระหว่าง	70-79	เกรด	B
คะแนนระหว่าง	80-100	เกรด	A

   โปรแกรมมีความสามารถดังนี้
   - บันทึกรายละเอียด รหัส, ชื่อนักศึกษา, คะแนนระหว่างภาค, คะแนนปลายภาค ลงไฟล์ข้อมูล

- รวมคะแนนจากแฟ้มข้อมูลเพื่อนำไปคำนวณตัดเกรด
- คำนวณการตัดเกรดตามเกณฑ์ที่กำหนดไว้
- คำนวณจำนวนคนที่ได้รับเกรดแต่ละเกรด
- แสดงรายงานผลรายละเอียดข้อมูลการตัดเกรดทั้งหมด พร้อมการสรุปจำนวนผู้ได้รับในแต่ละเกรด

โดยที่การเขียนโปรแกรมจะต้องสร้างเป็นฟังก์ชันแบ่งตามหน้าที่การทำงานในโปรแกรมให้เหมาะสม