ជំពូក ទី ១៣

ការប្រតិបត្តិទៅលើ **Data Files**

១. ការបើក និងបិទ File

File គឺជាការប្រមូលផ្តុំទិន្នន័យ ឬ សំណុំតួអក្សរ ឬ ជាអត្ថបទ ឬ កម្មវិធីមួយ។ តាមធម្មតា វាមាន files ពីរប្រភេទនៅក្នុងភាសា C++ គឺ ៖ sequential files និង random access files។ sequential files មានលក្ខណៈងាយស្រួលបង្កើតជាង random access files។ នៅក្នុង sequential files ទិន្នន័យ ឬ អត្ថបទត្រូវបានផ្ទុកឬអានបញ្ចេញមកវិញមានលក្ខណៈលំដាប់ លំដោយ។ នៅក្នុង random access files ទិន្នន័យអាចចូលប្រើនិងដំណើរការដោយមិនតាមលំដាប់លំដោយ។

fstream.h គឺជា header file ដែលផ្តល់នូវវិធីដំណើរការ input/output stream បានយ៉ាងល្អិតល្អន់ និងដែលអាចឲ្យគេប្រើសម្រាប់បញ្ចេញ-បញ្ចូលទិន្នន័យចំពោះលក្ខណៈពិសេសរបស់ភាសា ដូចជា classes, derived class, function overloading, virtual function និង multiple inheritance។ នៅក្នុងផ្នែកនេះ យើងនឹងសិក្សារបៀបបើក និងបិទ data file ដោយប្រើ header file ឈ្មោះ <fstream.h>។ បណ្តុំ class សម្រាប់បញ្ចេញ-បញ្ចូលទិន្នន័យរបស់ C++ បានប្រើ file I/O ច្រើនជាងវាប្រើការបញ្ចេញ-បញ្ចូលទិន្នន័យធម្មតា។ វិធីខាងក្រោមនេះត្រូវបានប្រើនៅក្នុង C++ សម្រាប់អាននិងសរសេរ file ៖

ifstream៖សម្រាប់អាន stream នៃ object ពី file មួយដែលបានកំណត់។

ofstream៖ សម្រាប់សរសេរ stream នៃ object ទៅលើ file មួយដែលបានកំណត់។

fstream៖ សម្រាប់អាននិងសរសេរ stream នៃ objects ទៅលើ file មួយដែលបានកំណត់។

header file ឈ្មោះ fstream.h គឺជា class ថ្មីមួយដែលមានមុខងារនិងទម្រង់ការប្រតិបត្តិ file សំខាន់ៗ។ fstream, ifstream និង ofstream ត្រូវបានចាត់ទុកដូចជា derived class ដោយសារ class objects ទាំងនេះបានបង្កើតរួចហើយនៅក្នុង class សម្រាប់បញ្ចេញ-បញ្ចូលទិន្នន័យសំខាន់ៗគឺឈ្មោះ <iostream.h>។

១.១ ការបើក file មួយ

ឧទាហរណ៍ខាងក្រោមនេះបង្ហាញពីរបៀបបើក អាន និងសរសេរទៅកាន់ files ជាភាសា C++។ Member function ឈ្មោះ open() ត្រូវបានប្រើសម្រាប់បង្កើត file pointer មួយ ដើម្បីបើក file មួយនៅក្នុង disk។

ក. ifstream**៖** header file ឈ្មោះ ifstream.h គឺជា derived class មួយមានប្រភពពី base class នៃ istream និងត្រូវបានប្រើសម្រាប់អាន stream នៃ objects ពី file មួយ។

ឧទាហរណ៍ code ខាងក្រោមបង្ហាញពីរបៀបបើក file មួយដើម្បីអានទិន្នន័យផ្ទុកក្នុង stream object ចេញពី file មួយ ដែលបានបញ្ចូល។

**#include <fstream.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**infile.open("data\_file");​ //** ការបើក file មួយឈ្មោះ data\_file

**...**

**...**

**}**

ខ. ofstream **៖** header file ឈ្មោះ ofstream.h មានប្រភពពី base class នៃ ostream ហើយត្រូវបានប្រើសម្រាប់សរសេរទិន្នន័យឬព័ត៌មាននៃ objects ទៅក្នុង file មួយ។

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញពីរបៀបបើក file មួយដើម្បីសរសេរទិន្នន័យឬ ព័ត៌មាននៃ objects ពី file មួយដែលបានបញ្ចូល។

**#include <fstream.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**outfile.open("data\_file"); //** ការបើក file មួយឈ្មោះ data\_file

**...**

**...**

**}**

គ. fstream**៖** header file ឈ្មោះ fstream.h គឺជា derived class មួយមានប្រភពពី base class នៃ iostream ហើយត្រូវបានប្រើសម្រាប់អាន និងសរសេរទិន្នន័យមួយនៃ objects ទៅក្នុង file មួយ។ include <fstream.h> បានបញ្ចូល header file ឈ្មោះ iostream.h ដោយស្វ័យប្រវត្ត។

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញនូវរបៀបបើក file មួយដើម្បីអាននិងសរសេរ ទិន្នន័យមួយនៃ object ពី file មួយដែលបានបញ្ចូល។

**#include <fstream.h>**

**void main(){**

**fstream infile; //** ការប្រកាស object នៃ fstream ឈ្មោះ infile

**infile.open("data\_file", ios::in | ios::out);​**

**//** ការបើក file មួយឈ្មោះ data\_file

**...**

**...**

**}**

នៅពេល file មួយបានបើក ដើម្បីអាននិងសរសេរនោះ I/O streams រក្សាដាននៃ file pointers ពីរ គឺមួយសម្រាប់ប្រតិបត្តិការបញ្ចូលទិន្នន័យនិងមួយទៀតសម្រាប់ប្រតិបត្តិការ បញ្ចេញទិន្នន័យ។

ទម្រង់ទូទៅ សម្រាប់ការបើក file មួយគឺ ៖

*void ifstream::open( const char\* fname,*

*int m= ios::in,*

*int port=filebuf::openport);*

*void ofstream::open( const char\* fname,*

*int m= ios::out,*

*int port=filebuf::openport);*

*void fstream::open( const char\* fname,*

*int m,*

*int port=filebuf::openport);*

ចូរកត់សម្គាល់ថា ចំពោះ instance នៃ istream (input) នេះនៅក្នុង default mode គឺ ios::in ហើយចំពោះ instance នៃ ofstream (input) នោះ default mode របស់វាគឺ ios::out។ ទោះជាយ៉ាងណាក៏ដោយ ចំពោះ instance នៃ fstream (input/output) វិញពុំមាន default mode ឡើយ។ សញ្ញាណនព្វន្តលេខ ឈ្នាប់ OR ត្រូវបានប្រើសម្រាប់ប្រកាស mode លើសពីមួយ។ ការរាយឈ្មោះ member functions ខាងក្រោមនេះត្រូវបានប្រើជា file attributes សម្រាប់ប្រតិបត្តិការបើក file ប្រភេទផ្សេងៗគ្នា។

|  |  |
| --- | --- |
| ឈ្មោះរបស់ functions | អត្ថន័យ |
| ios::in  ios::out  ios::app  ios::ate  ios::trunc  ios::nocreate  ios::noreplace  ios::binary | - បើក file មួយសម្រាប់អាន  - បើក file មួយសម្រាប់សរសេរ  - បើក file មួយសម្រាប់ភ្ជាប់ទៅផ្នែកខាងចុងនៃ file មួយ  - នាំទៅរកផ្នែកខាងចុងនៃ file មួយ  - តម្រឹម file នៅពេលបើក  - មិនត្រូវព្យាយាមបង្កើត file ឡើយបើសិនជា file នោះមិនមាន  - បើក file មិនបានជោគជ័យ បើសិនជា file នោះមានហើយ  - បើក file មួយសម្រាប់ binary mode រីឯ default គឺជា text mode |

ចំពោះ argument ទីបី នៅក្នុង Borland C++ មានតម្លៃដូចនៅក្នុងតារាងខាងក្រោម ៖

|  |  |
| --- | --- |
| តម្លៃ | អត្ថន័យ |
| 0  1  2  4  8 | defaultread only filehidden filesystem filearchive file |

១.២ ការបិទ file មួយ

member function ឈ្មោះ close() ត្រូវបានប្រើសម្រាប់បិទ file មួយដែលបានបើកដើម្បីដំណើរការអាន, សរសេរ ឬ អាននិងសរសេរ។ member function close() នេះត្រូវបានហៅប្រើដោយស្វ័យប្រវត្ត តាមរយៈ destructor functions។ ទោះជាយ៉ាងណាក៏ដោយ គេអាចហៅ អនុគមន៍នេះមកប្រើ ដើម្បីបិទ file តាមលក្ខណៈបញ្ជាក់ឲ្យ។ វាពុំមានការផ្ទុក arguments ណាមួយឡើយ។

ទម្រង់ទូទៅនៃ member function close() គឺ ៖

**#include <fstream.h>**

**void main(){**

**fstream infile; //** ការប្រកាស object នៃ fstream ឈ្មោះ infile

**infile.open("data\_file", ios::in||ios::out);**

**. . . . .**

**. . . . .**

**infile.close(); //** ការប្រើអនុគមន៍ close() របស់ infile

**}**

២. Member functions ដែលមានសភាពជា stream

នៅក្នុងភាសា C++, file stream classes បានទទួលលក្ខណៈ member ដែលមានសភាពជា stream ពី ios class។ member functions មានសភាពជា stream នេះផ្តល់នូវព័ត៌មាន ដូចជា ទីបញ្ចប់នៃ file បានទៅដល់ ឬការបើក file ពុំបានជោគជ័យ -ល-។ member functions មានសភាពជា stream នៅខាងក្រោមនេះត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើល ការបើក file ពុំបានជោគជ័យណាមួយដែលកើតឡើង នៅពេលគេព្យាយាមបើក file មួយពី diskette។

ក.eof()**៖** member function នេះត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើលថាតើ file pointer បានទៅដល់ទីបញ្ចប់នៃ file ហើយឬនៅ។ បើសិនជាវាបានទៅដល់ចុងបញ្ចប់នៃ file នោះ member function eof() ឲ្យតម្លៃ 1 ឬមិនសូន្យ ផ្ទុយទៅវិញ member function eof() ឲ្យតម្លៃ 0។

ទំរង់ទូទៅនៃ member function eof() គឺ ៖

**#include <fstream.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ fstream ឈ្មោះ infile

**infile.open("text"); //** ការបើក file មួយឈ្មោះ text

**while(!infile.eof()){**

**. . . . .**

**. . . . .**

**}**

**}**

ខ. fail()**៖** member function មានសភាពជា stream នេះត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើលថាតើ file មួយបានបើកសម្រាប់ input ឬ output ដោយជោគជ័យឬទេ ឬការប្រតិបត្តិមិនត្រឹមត្រូវណាមួយ ឬវាមាន error ណាមួយមិនអាចបំបាត់បាន។ បើសិនជាវាប្រតិបត្តិមិនបានជោគជ័យនោះ វានឹងឲ្យតម្លៃពិត (ឬ មិនសូន្យ)។

ទំរង់ទូទៅនៃ member function fail() គឺ ៖

**#include <fstream.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**infile.open("text"); //** ការបើក file មួយឈ្មោះ text

**while(infile.fail()){**

**cout << " Could't open a file." << endl;**

**continue;**

**. . . . .**

**}**

**}**

គ. bad()**៖** member function មានសភាពជា stream នេះត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើលថាតើការប្រតិបត្តិ file មិនត្រឹមត្រូវណាមួយបានកើតឡើង ឬវាមាន error មួយកើតឡើងដែលមិនអាចបំបាត់បាន។ bad() function នេះឲ្យតម្លៃមិនសូន្យបើសិនជាវាអាចបំបាត់ error មកវិញពីការប្រតិបត្តិណាមួយ ផ្ទុយទៅវិញវាឲ្យតម្លៃសូន្យ។

ទំរង់ទូទៅនៃ member function bad() គឺ ៖

**#include <fstream.h>**

**#include <stdlib.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**infile.open("text"); //** ការបើក file មួយឈ្មោះ text

**if(infile.bad()){**

**cerr << "Open failure" << endl;**

**exit(1);**

**}**

**. . . . .**

**}**

ឃ. good()**៖** member function មានសភាពជា stream នេះត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើលថា តើការប្រតិបត្តិ file មុននេះបានជោគជ័យឬមិន។ good() នេះឲ្យតម្លៃមិនសូន្យ បើសិនជាគ្មាន error កើតឡើង។ នេះមានន័យថា គ្រប់ functions ទាំងអស់ខាងលើ (eof(), fail() និង bad()) មិនពិត។ ឧទាហរណ៍ បើ infile.good() ពិតនោះគ្រប់ functions ទាំងអស់ប្រតិបត្តិបានប្រសើរជាមួយ stream file ហើយគេអាចដំណើរការប្រតិបត្តិបញ្ចូល-បញ្ចេញទិន្នន័យបាន។ នៅពេលវាឲ្យតម្លៃមិនពិតនោះពុំមានការប្រតិបត្តិណាមួយអាចដំណើរការបានឡើយ។

ទម្រង់ទូទៅនៃ member function good() គឺ ៖

**#include <fstream.h>**

**#include <stdlib.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**infile.open("text"); //** ការបើក file មួយឈ្មោះ text

**if(infile.good()){**

**. . . . .**

**. . . . .**

**}**

**​ }**

៣. ការអាននិងសរសេរតួអក្សរពី file មួយ

member functions ខាងក្រោមនេះត្រូវបានប្រើសម្រាប់អាននិងសរសេរមួយតួអក្សរពី

file មួយ ដែលបានបញ្ជាក់។

ក. get()**៖** អនុគមន៍ get() ត្រូវបានប្រើសម្រាប់អានមួយតួអក្សរពី file ដែលបានបញ្ជាក់។

ទម្រង់ទូទៅនៃ member function get() គឺ ៖

**#include <fstream.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**char ch; //** ការប្រកាសអញ្ញាត ch ជាប្រភេទ char

**infile.open("text"); //** ការបើក file មួយឈ្មោះ text

**. . . . .**

**. . . . .**

**while(!infile.eof()){**

**ch = infile.get(); //** ការហៅអនុគមន៍ get() តាមរយៈ infile object

**. . . . .**

**. . . . .**

**} // end of while loop**

**}**

ខ. put()**៖** អនុគមន៍ put() ត្រូវបានប្រើសម្រាប់សរសេរតួអក្សរមួយទៅកាន់ file មួយឬ output stream មួយដែលបានបញ្ជាក់។

ទម្រង់ទូទៅនៃ member function put() គឺ ៖

**#include <fstream.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**char ch; //** ការប្រកាសអញ្ញាត ch ជាប្រភេទ char

**outfile.open("text"); //** ការបើក file មួយឈ្មោះ text

**. . . . .**

**. . . . .**

**while((ch=cin.get())!=EOF){**

**outfile.put(ch); //** ការហៅអនុគមន៍ put() តាមរយៈ outfile object

**. . . . .**

**. . . . .**

**} // end of while loop**

**}**

ឧទាហរណ៍ទី ១ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញការសរសេរ file ជាច្រើនបន្ទាត់ដោយបានកំណត់ឈ្មោះថា "text"។

**// storing a text on a file**

**#include <fstream.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**outfile.open("text"); //** ការបើក file មួយឈ្មោះ text

**//** code ខាងក្រោមនេះសរសេរព័ត៌មានមួយចំនួនចូលទៅក្នុង file ឈ្មោះ text

**outfile << "This is a test \n";**

**outfile << " program to store \n";**

**outfile << " a set of lines on to a file.\n";**

**outfile.close(); //** ការបិទ file

**}**

ឧទាហរណ៍ទី ២ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញការសរសេរ file ជាច្រើនបន្ទាត់ដែលឈ្មោះរបស់វាត្រូវបានបញ្ចូលដោយអ្នកប្រើប្រាស់។

**// storing a text on a specified file**

**#include <fstream.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជាប្រភេទ char array ចំនួន 10តួអក្សរ

**cout << "Enter a file name to be opened :\n";**

**cin >> fname; //** ការបញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**outfile.open(fname); //** ការបើក file មួយតាមរយៈ fname

**//** code ខាងក្រោមនេះសរសេរព័ត៌មានមួយចំនួនចូលទៅក្នុង file មួយ

**outfile << "This is a test \n";**

**outfile << " program to store \n";**

**outfile << " a set of lines on to a file \n";**

**outfile.close(); //** ការបិទ file

**}**

ឧទាហរណ៍ទី ៣ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញនូវការអានបញ្ចូលទិន្នន័យជាច្រើនបន្ទាត់ពី keyboard ហើយផ្ទុកនៅលើ file មួយដែលបានកំណត់។

**// reading a text and store it on a specified file**

**#include <fstream.h>**

**#define MAX 2000**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**char fname[10], line[MAX];//** ប្រកាសអញ្ញាត fname និង line ជាប្រភេទ char array

**cout << "Enter a file name to be opened :\n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**outfile.open(fname); //** ការបើក file មួយតាមរយៈ fname

**cout <<"Enter a set of lines and terminate with @:\n";**

**cin.get(line, MAX, '@');**

**//** បញ្ចូលតម្លៃឲ្យ line តាមរយៈ keyboard រហូតដល់ជួបតួអក្សរ ‘@’

**cout << "Given input :\n"; //** បង្ហាញអក្សរ Given input : មកលើអេក្រង់

**cout << line; //** បង្ហាញតម្លៃរបស់ line មកលើអេក្រង់

**cout << " storing onto a file ... \n";**

**outfile << line; //** សរសេរបញ្ចូលតម្លៃរបស់ line ទៅក្នុង outfile

**outfile.close(); //** ការបិទ file

**}**

ឧទាហរណ៍ទី ៤ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញនូវរបៀបអាន text file ហើយបង្ហាញខ្លឹមសារនៅលើអេក្រង់។

**// reading and displaying a file**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <stdlib.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**char fname[10]; //** ប្រកាសអញ្ញាត fname ជាប្រភេទ char array មានចំនួន 10 តួអក្សរ

**char ch; //** ប្រកាសអញ្ញាត ch ជាប្រភេទ char

**cout << "Enter a file name :\n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**infile.open(fname); //** ការបើក file មួយតាមរយៈ fname

**if (infile.fail()){**

**cerr << "No such a file exists.\n";**

**exit(1);**

**}**

**while(!infile.eof()){**

**ch = (char) infile.get(); //** ទទួលតម្លៃពី infile ម្តងមួយតួអក្សរ

**cout.put(ch); //** បង្ហាញតម្លៃរបស់ ch មកលើអេក្រង់

**}**

**infile.close(); //** ការបិទ file

**} // end of main**

ឧទាហរណ៍ទី ៥ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញនូវការចម្លងខ្លឹមសារនៃ text file មួយទៅកាន់ file មួយទៀត។

**// file copy**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <stdlib.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**char fname1[10], fname2[10];**

**//** ប្រកាសអញ្ញាត fname1 និង fname2 ជាប្រភេទ char array មានចំនួន 10តួអក្សរ

**char ch; //** ប្រកាសអញ្ញាត ch ជាប្រភេទ char

**cout << "Enter a file name to be copied :\n";**

**cin >> fname1; //** បញ្ចូលតម្លៃឲ្យ fname1 តាមរយៈ keyboard

**cout << "New file name :\n";**

**cin >> fname2; //** បញ្ចូលតម្លៃឲ្យ fname2 តាមរយៈ keyboard

**infile.open(fname1); //** ការបើក file មួយតាមរយៈ fname1

**if (infile.fail()){**

**cerr << "No such a file exists \n";**

**exit(1);**

**}**

**outfile.open(fname2); //** ការបើក file មួយតាមរយៈ fname2

**if (outfile.fail()){**

**cerr << "Unable to create a file.\n";**

**exit(1);**

**}**

**while(!infile.eof()) {**

**ch = (char) infile.get(); //** ទទួលតម្លៃពី infile ម្តងមួយតួអក្សរ

**outfile.put(ch); //** បញ្ចូលតម្លៃរបស់ ch ទៅក្នុង outfile

**}**

**infile.close(); //** ការបិទ infile

**outfile.close(); //** ការបិទ outfile

**} // end of main**

ឧទាហរណ៍ទី ៦ ៖ កម្មវិធីខាងក្រោមនេះធ្វើការលុបតំណកឃ្លា ដូចជា horizontal tab, vertical tab, space, line feed, new line និង carriage return ពី text file មួយ ហើយផ្ទុកខ្លឹមសាររបស់ file ដែលគ្មានតំណកឃ្លានៅលើ file មួយផ្សេងទៀត។

**// deleting white spaces from a file**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <stdlib.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**char fname1[10], fname2[10];**

**char ch; //** ប្រកាសអញ្ញាត ch ជាប្រភេទ char

**cout << "Enter a file name to be copied :\n";**

**cin >> fname1; //** បញ្ចូលតម្លៃឲ្យ fname1 តាមរយៈ keyboard

**cout << "New file name :\n";**

**cin >> fname2; //** បញ្ចូលតម្លៃឲ្យ fname2 តាមរយៈ keyboard

**infile.open(fname1); //** ការបើក file មួយតាមរយៈ fname1

**if (infile.fail()){**

**cerr << "No such a file exists.\n";**

**exit(1);**

**}**

**outfile.open(fname2); //** ការបើក file មួយតាមរយៈ fname2

**if (outfile.fail()){**

**cout << "Unable to create a file.\n";**

**exit(1);**

**}**

**while(!infile.eof()) {**

**ch = (char) infile.get(); //** ទទួលតម្លៃពី infile ម្តងមួយតួអក្សរ

**if (ch == ' ' || ch == '\t' || ch == '\n');**

**else**

**outfile.put(ch); //** បញ្ចូលតម្លៃរបស់ ch ទៅក្នុង outfile

**}**

**infile.close(); //** ការបិទ infile

**outfile.close(); //** ការបិទ outfile

**} // end of main**

ឧទាហរណ៍ទី ៧ ៖ កម្មវិធីខាងក្រោមនេះធ្វើការបំលែងតួអក្សរជាអក្សរតូច (lower case) ទៅជាតួអក្សរធំ (upper case) ពី text file មួយ។

**// converting a lower case to upper case letters**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**ifstream infile; //** ការប្រកាស object នៃ ifstream ឈ្មោះ infile

**char fname1[10], fname2[10];**

**char ch, uch; //** ប្រកាសអញ្ញាត ch, uch ជាប្រភេទ char

**cout << "Enter a file name to be copied :\n";**

**cin >> fname1; //** បញ្ចូលតម្លៃឲ្យ fname1 តាមរយៈ keyboard

**cout << "New file name :\n";**

**cin >> fname2; //** បញ្ចូលតម្លៃឲ្យ fname2 តាមរយៈ keyboard

**infile.open(fname1); //** ការបើក file មួយតាមរយៈ fname1

**if (infile.fail()){**

**cerr << "No such a file exists.\n";**

**exit(1);**

**}**

**outfile.open(fname2); //** ការបើក file មួយតាមរយៈ fname2

**if (outfile.fail()){**

**cout << "Unable to create a file.\n";**

**exit(1);**

**}**

**while(!infile.eof()){**

**ch = (char) infile.get(); //** ទទួលតម្លៃពី infile ម្តងមួយតួអក្សរ

**uch = toupper(ch); //** ហៅអនុគមន៍ toupper() មកប្រើប្តូរអក្សរតូចទៅជាអក្សរធំ

**outfile.put(uch); //** បញ្ចូលតម្លៃរបស់ ch ទៅក្នុង outfile

**}**

**infile.close(); //** ការបិទ infile

**outfile.close(); //** ការបិទ outfile

**} // end of main**

៤. ការប្រតិបត្តិទៅលើ Binary File

នៅក្នុង C++ តាមធម្មតា ការប្រតិបត្តិ file stream ត្រូវបានធ្វើឡើងនៅក្នុង text mode ក៏ប៉ុន្តែវាផ្តល់ការប្រតិបត្តិជា binary file ដែរ។ binary file គឺជា file ដែលអាចប្រើតាមលំដាប់លំដោយ ហើយទិន្នន័យត្រូវបានផ្ទុកនិងអានចេញមកវិញបន្តបន្ទាប់គ្នាក្នុងទម្រង់ជា binary ជំនួសឲ្យតួអក្សរ ASCII។ ឧទាហរណ៍ binary file មានចំនួនគត់, ចំនួនទសភាគ, array នៃ structures -ល-។ ដំណើរការ binary file មានប្រយោជន៍ចំពោះការរៀបចំនិងបង្កើតនូវទិន្នន័យសាំញ៉ាំ ឬអាននិងសរសេរព័ត៌មានជា binary។

Text file បង្កើតឡើងដោយ C++ អាចសរសេរតាមរយៈ editor ធម្មតា ឬតាមរយៈ word processor។ Text file អាចបញ្ជូនពីប្រព័ន្ធកុំព្យូទ័រមួយទៅប្រព័ន្ធកុំព្យូទ័រមួយទៀតបានយ៉ាង ងាយ។ ផ្ទុយទៅវិញ binary file មានលក្ខណៈសុក្រិតជាងចំពោះលេខ ព្រោះវាផ្ទុកនូវការតាងតម្លៃនៅខាងក្នុងយ៉ាងជាក់លាក់។ វាពុំមានការបំលែង errors ឬផ្តុំ errors ឡើយ។ ការរក្សាទិន្នន័យក្នុងទម្រង់ជា binary មានលក្ខណៈលឿនជាង ព្រោះវាពុំមានធ្វើការបំលែងនៅពេលផ្ទុកទិន្នន័យទៅក្នុង file។ ជាធម្មតា file ដែលមានទិន្នន័យទម្រង់ជា binary ត្រូវការទីតាំងផ្ទុកតូចជាង។ ទោះជាយ៉ាងណាក៏ដោយ file ដែលមានទិន្នន័យ ទំរង់ជា binary មិនអាចបញ្ជូនពីប្រព័ន្ធកុំព្យូទ័រមួយទៅកាន់ប្រព័ន្ធកុំព្យូទ័រមួយទៀតបានយ៉ាងងាយឡើយ អាស្រ័យដោយភាពខុសគ្នាពីកុំព្យូទ័រមួយទៅកុំព្យូទ័រមួយទៀត នៅក្នុងការតាងទិន្នន័យផ្នែកខាងក្នុងរបស់វា។

ដើម្បីបើក binary file មួយវាត្រូវការប្រើ mode ខាងក្រោមនេះ ៖

infile("data", ios:: binary);

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញរបៀបបើក binary file មួយជាភាសា C++។

**#include <fstream.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**outfile("data", ios:: binary); //** ការបើក file មួយឈ្មោះ data

**.....**

**.....**

**}**

ឧទាហរណ៍ទី ១ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការបើក binary file មួយដើម្បីផ្ទុកសំណុំចំនួននៅលើ file មួយដែលបានកំណត់។

**// storing data on a file using binary file operations**

**#include <fstream.h>**

**#include <iostream.h>**

**void main(){**

**ofstream outfile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ outfile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**float x, y, temp; //** ការប្រកាសអញ្ញាត x, y, temp ជាប្រភេទ float

**cout << "Enter a file name :\n";**

**//** ការបង្ហាញអក្សរ Enter a file name : មកលើអេក្រង់

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**outfile.open(fname, ios::out || ios::binary);**

**x = 1.5; //** ការកំណត់តម្លៃ 1.5 ទៅឲ្យអញ្ញាត x

**y = 10.5; //** ការកំណត់តម្លៃ 10.5 ទៅឲ្យអញ្ញាត y

**cout << "x temp " << endl;**

**cout << "-----------------" << endl;**

**while(x <= y){**

**temp = x \* x; //** ការធ្វើប្រមាណវិធីគុណ x និង x រួចកំណត់តម្លៃទៅឲ្យអញ្ញាត temp

**outfile << x << '\t' << temp << endl;**

**cout << x << '\t' << temp << endl;**

**x = x + 1.5; //** ការបង្កើនតម្លៃ 1.5 ទៅឲ្យអញ្ញាត x

**}**

**outfile.close(); //** ការបិទ outfile

**} // end of main**

ឧទាហរណ៍ទី ២ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការបើក binary file មួយ ដើម្បីអានសំណុំចំនួនរហូតដល់ជួបសញ្ញាទីបញ្ចប់នៃ file រួចហើយបង្ហាញខ្លឹមសារនៃ file នៅលើអេក្រង់។

**// reading a data from the file using binary file operations**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <iomanip.h>**

**void main(){**

**ifstream infile; //** ការប្រកាស object នៃ ofstream ឈ្មោះ infile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**float x, y, temp; //** ការប្រកាសអញ្ញាត x, y, temp ជាប្រភេទ float

**cout << "Enter a file name :\n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**infile.open(fname, ios::in || ios::binary);**

**//** ទាញយកតម្លៃចេញពី file ទៅឲ្យអញ្ញាត x

**cout << "x temp " << endl;**

**cout << "---------------" << endl;**

**while(!infile.eof()){**

**infile >> x >> '\t' >> temp;**

**cout << x << '\t' << temp << endl;**

**}**

**infile.close(); //** ការបិទ infile

**} // end of main**

៥. ការប្រើ Structures ជាមួយនឹង File

យើងបានដឹងហើយថា structure មួយគឺជាប្រភេទទិន្នន័យដែលកំណត់ដោយអ្នកប្រើប្រាស់ ហើយធាតុរបស់វាមានប្រភេទទិន្នន័យមិនដូចគ្នា។ នៅក្នុងផ្នែកនេះ យើងលើកយកទិន្នន័យប្រភេទ structure ដែលអាចអាននិងសរសេរតាមរយៈ file មួយដែលបានកំណត់។ array នៃ structures អាចផ្ទុកនិងចូលប្រើបានដោយសារការប្រើពាក្យបញ្ជារបស់ file។ ជួនកាល វាត្រូវការផ្ទុកធាតុនៃ structure ដែលនៅផ្តុំគ្នា ហើយទាញយកវាមកវិញក្នុងទម្រង់ប្រហាក់ប្រហែលគ្នា។

ឧទាហរណ៍ខាងក្រោមនេះបង្ហាញនូវរបៀបដែល file មួយត្រូវបើកឡើង ដើម្បីអាននិងសរសេរប្រភេទទិន្នន័យជា structure។

**#include <fstream.h>**

**struct School {**

**char name[20]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 20តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int

**};**

**void main(){**

**School student; //** ការប្រកាស object នៃ school ឈ្មោះ student

**fstream myfile; //** ការប្រកាស object នៃ fstream ឈ្មោះ myfile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**myfile.open(fname, ios::in || ios::out);**

**.....**

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << " storing onto the file ....\n";**

**myfile << student.name<< setw(5)<< student.age<<endl;**

**.....**

**//** ការបើក file និងបញ្ចូលតម្លៃទៅក្នុង file

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << " reading from the file ....\n";**

**while(!myfile.eof()){**

**myfile >> student.name >> setw(5) >> student.age;**

**.....**

**}**

**myfile.close(); //** ការបិទ myfile

**} // end of main**

ឧទាហរណ៍ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការអានទិន្នន័យចំពោះធាតុនៃ structure ដូចជា name, age, sex, height និង weight ពី keyboard ហើយផ្ទុកវានៅលើ file មួយដែលបាន កំណត់។ បន្ទាប់មក file ដដែលនេះត្រូវបានបើកដើម្បីអាន និងបង្ហាញខ្លឹមសារនៃ file នៅលើអេក្រង់។

**// structure and file operations**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <iomanip.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**#define MAX 10**

**struct School {**

**char name[10]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char

**float height, weight; //** ការប្រកាសអញ្ញាត height, weight ជាប្រភេទ float

**};**

**void main(){**

**School student[MAX]; //** ការប្រកាស object នៃ school ឈ្មោះ student ជា array

**fstream infile; //** ការប្រកាស object នៃ fstream ឈ្មោះ infile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**int i, n; //** ការប្រកាសអញ្ញាត i, n ជាប្រភេទ int

**cout << "Enter a file name to be stored :\n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**infile.open(fname, ios::in || ios::out);**

**// reading from the keyboard**

**cout << "How many records are to be stored :\n";**

**cin >> n; //** បញ្ចូលតម្លៃឲ្យ n តាមរយៈ keyboard

**cout << "Enter the following information : \n";**

**for (i = 0; i < n; i++){**

**cout << "Name : "; //** ការបង្ហាញអក្សរ Name : មកលើអេក្រង់

**cin >> student[i].name; //** បញ្ចូលតម្លៃឲ្យអញ្ញាត name តាមរយៈ keyboard

**cout << "Age : "; //** ការបង្ហាញអក្សរ Age : មកលើអេក្រង់

**cin >> student[i].age; //** បញ្ចូលតម្លៃឲ្យអញ្ញាត age តាមរយៈ keyboard

**cout << "Sex : "; //** ការបង្ហាញអក្សរ Sex : មកលើអេក្រង់

**cin >> student[i].sex; //** បញ្ចូលតម្លៃឲ្យអញ្ញាត sex តាមរយៈ keyboard

**cout << "Height : "; //** ការបង្ហាញអក្សរ Height : មកលើអេក្រង់

**cin >> student[i].height;//** បញ្ចូលតម្លៃឲ្យអញ្ញាត height តាមរយៈ keyboard

**cout << "Weight : "; //** ការបង្ហាញអក្សរ Weight : មកលើអេក្រង់

**cin >> student[i].weight;//** បញ្ចូលតម្លៃឲ្យអញ្ញាត weight តាមរយៈ keyboard

**}**

**//** ផ្ទុកតម្លៃនៅក្នុង file មួយ

**infile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << " storing onto the file ... \n";**

**for (i = 0; i < n; i++){**

**infile << student[i].name<< setw(5)<<student[i].age**

**<< setw(10) << student[i].sex << setw(5) <<**

**student[i].height << setw(5) << student[i].weight**

**<< endl;**

**}**

**infile.close(); //** ការបិទ infile

**//** ទាញយកតម្លៃចេញពី file

**infile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ... \n";**

**i = 0; //** ការកំណត់តម្លៃ 0 ទៅឲ្យអញ្ញាត i

**while (!infile.eof()){**

**infile >> student[i].name>> setw(5)>>student[i].age**

**>> setw(10) >> student[i].sex >> setw(5) >>**

**student[i].height >> setw(5) >> student[i].weight;**

**i++; //** ការបង្កើនតម្លៃមួយឯកតាទៅឲ្យអញ្ញាត i

**}**

**for (int j = 0; i < n; j++){**

**cout << student[j].name<< setw(5)<< student[j].age**

**<< setw(10) << student[j].sex << setw(5) <<**

**student[j].height << setw(5) << student[j].weight**

**<< endl;**

**}**

**infile.close(); //** ការបិទ infile

**} // end of main**

៦. ការប្រើ Classes ជាមួយនឹង File

ដោយសារ C++ ជាភាសាបែប OOP នោះ វាជាមូលហេតុដែលត្រូវសិក្សាពី objects ដែលអាចអាន និងសរសេរទៅលើ file មួយ។ នៅក្នុងផ្នែកនេះនឹងបង្ហាញឲ្យឃើញនូវរបៀប objects ដែលអាចសរសេរ និងអានពីឧបករណ៍ខាងក្រៅតាមធម្មតាគឺ disk។ header file ឈ្មោះ fstream.h ត្រូវតែដាក់បញ្ចូលសម្រាប់ប្រើការប្រតិបត្តិបញ្ចោញ-បញ្ចូលទិន្នន័យរបស់ file។ mode នៃការប្រតិបត្តិលើ file ដូចជា អាន, សរសេរ ឬអាននិងសរសេរត្រូវតែកំណត់បញ្ជាក់ឲ្យបានច្បាស់។ ការប្រតិបត្តិលើ binary file តម្រូវការអនុវត្តនូវលក្ខណៈបញ្ចោញ-បញ្ចូលទិន្នន័យដោយប្រើ member functions get() និង put() សម្រាប់ការប្រតិបត្តិសៀតបញ្ចូលទិន្នន័យ និងការទាញយកទិន្នន័យ។

member functions read() និង write() ត្រូវបានប្រើសម្រាប់អាន និងសរសេរ stream មួយនៃ objects ពី file ដែលបានកំណត់បញ្ជាក់។

***ក****.* ***ការអាន*** *object* ***មួយចេញពី*** *file***៖** member function read() ត្រូវបានប្រើសម្រាប់ទាញយកទិន្នន័យ ចំពោះ stream នៃ object ពី file មួយដែលបានកំណត់។

ទំរង់ទូទៅនៃ member function read() គឺ ៖

*infile.read ((char \*) &obj, sizeof(obj));*

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញពីរបៀបអាន objects នៃ class មួយពី file ដោយប្រើ read() member function ៖

**// reading an object from a file**

**#include <fstream.h>**

**class Student {**

**protected :**

**char name[20]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 20តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char

**public :**

**void getdata(); //** ការប្រកាសអនុគមន៍ getdata()

**void display(); //** ការប្រកាសអនុគមន៍ display()

**};**

**void main(){**

**Student obj; //** ការបង្កើត object នៃ Student ឈ្មោះ obj

**fstream infile; //** ការបង្កើត object នៃ fstream ឈ្មោះ infile

**infile.open("data", ios::in); //** ការបើក file មួយឈ្មោះ data

**infile.read((char\*) &obj, sizeof(obj));**

**.....**

**.....**

**infile.close(); //** ការបិទ infile

**}**

***ខ****.* ***ការសរសេរ*** *object* ***មួយទៅក្នុង*** *file***៖** member function write() ត្រូវបានប្រើសម្រាប់រក្សាទុក stream នៃ objects ទៅក្នុង file មួយដែលបានកំណត់។

ទម្រង់ទូទៅនៃ member function write() គឺ ៖

*infile.write((char \*) &obj, sizeof(obj));*

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញពីរបៀបសរសេរ object មួយទៅក្នុង file ដោយប្រើ write() member function ៖

**// writing an object on a file**

**#include <fstream.h>**

**class Student {**

**protected :**

**char name[20]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 20តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char

**public :**

**void getdata(); //** ការប្រកាសអនុគមន៍ getdata()

**void display(); //** ការប្រកាសអនុគមន៍ display()

**};**

**void main(){**

**Student obj; //** ការបង្កើត object នៃ Student ឈ្មោះ obj

**fstream outfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ outfile

**outfile.open("data", ios::out); //** ការបើក file មួយឈ្មោះ data

**outfile.write((char\*) &obj, sizeof(obj));**

**.....**

**.....**

**outfile.close(); //** ការបិទ outfile

**}**

ឧទាហរណ៍ទី ១ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការអាន class object នៃ Student ដូចជា name, age, sex, height និង weight ពី keyboard ហើយផ្ទុកវានៅលើ file មួយដែលបានកំណត់ដោយប្រើ member function read() និង write()។ បន្ទាប់មក file ដដែលនេះត្រូវបានបើកដើម្បីអាន និងបង្ហាញខ្លឹមសារនៃ file នៅលើអេក្រង់។

**// classes and file operations using read and write**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <iomanip.h>**

**class Student {**

**protected :**

**char name[10]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int មានលក្ខណៈ protected

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char មានលក្ខណៈ protected

**float height, weight; //** ការប្រកាសអញ្ញាត height, weight ជាប្រភេទ float

**public :**

**void getdata(); //** ការប្រកាសអនុគមន៍ getdata() មានលក្ខណៈ public

**void display(); //** ការប្រកាសអនុគមន៍ display() មានលក្ខណៈ public

**};**

**void Student::getdata(){**

**cout << "Enter the following information :\n";**

**cout << "Name : "; //** ការបង្ហាញអក្សរ Name : មកលើអេក្រង់

**cin >> name; //** បញ្ចូលតម្លៃឲ្យ name តាមរយៈ keyboard

**cout << "Age : "; //** ការបង្ហាញអក្សរ Age : មកលើអេក្រង់

**cin >> age; //** បញ្ចូលតម្លៃឲ្យ age តាមរយៈ keyboard

**cout << "Sex : "; //** ការបង្ហាញអក្សរ Sex : មកលើអេក្រង់

**cin >> sex; //** បញ្ចូលតម្លៃឲ្យ sex តាមរយៈ keyboard

**cout << "Height : "; //** ការបង្ហាញអក្សរ Height : មកលើអេក្រង់

**cin >> height; //** បញ្ចូលតម្លៃឲ្យ height តាមរយៈ keyboard

**cout << "Weight : "; //** ការបង្ហាញអក្សរ Weight : មកលើអេក្រង់

**cin >> weight; //** បញ្ចូលតម្លៃឲ្យ weight តាមរយៈ keyboard

**}**

**void Student::display(){**

**cout << name << setw(5) << age << setw(10) << sex**

**<< setw(5)<< height<< setw(5) << weight<< endl;**

**}**

**void main(){**

**Student obj; //** ការបង្កើត object នៃ Student ឈ្មោះ obj

**fstream myfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ myfile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**cout << "Enter a file name to be stored :\n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**myfile.open(fname, ios::in || ios::out);**

**// reading from the keyboard**

**obj.getdata(); //** ការហៅអនុគមន៍ getdata() មកប្រើតាមរយៈ obj នៃ Student class

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << "Storing on the file ... \n";**

**myfile.write((char\*) &obj, sizeof(obj));**

**myfile.close(); //** ការបិទ myfile

**// reading from the file**

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ...\n";**

**myfile.read((char\*) &obj, sizeof(obj));**

**obj.display(); //** ការហៅអនុគមន៍ display() មកប្រើតាមរយៈ obj នៃ Student class

**myfile.close(); //** ការបិទ myfile

**}**

ឧទាហរណ៍ទី ២ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការអានទិន្នន័យចំពោះធាតុនៃ structure ដូចជា name, age, sex, height និង weight ពី keyboard ហើយផ្ទុកវានៅលើ file មួយដែលបានកំណត់ដោយប្រើ member function read() និង write()។ បន្ទាប់មក file ដដែលនេះត្រូវបានបើកដើម្បីអាន និងបង្ហាញខ្លឹមសារនៃ file នៅលើអេក្រង់។

**// structures and file operations using read and write**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <iomanip.h>**

**#include <stdlib.h>**

**#include <ctype.h>**

**#define MAX 10**

**struct School {**

**char name[10]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char

**float height, weight; //** ការប្រកាសអញ្ញាត height, weight ជាប្រភេទ float

**};**

**void main(){**

**School student[MAX]; //** ការបង្កើត object student នៃ School ចំនួន MAX

**fstream myfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ myfile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**int i, n; //** ការប្រកាសអញ្ញាត i, n ជាប្រភេទ int

**cout << "Enter a file name to be stored ? \n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**myfile.open(fname, ios::in || ios::out);**

**// reading from the keyboard**

**cout << "How many records are to be stored ? \n";**

**cin >> n; //** បញ្ចូលតម្លៃឲ្យ n តាមរយៈ keyboard

**cout << "Enter the following information \n";**

**for (i = 0; i < n; i++){**

**cout << "Name : "; //** ការបង្ហាញអក្សរ Name : មកលើអេក្រង់

**cin >> student[i].name; //** បញ្ចូលតម្លៃឲ្យ name តាមរយៈ keyboard

**cout << "Age : "; //** ការបង្ហាញអក្សរ Age : មកលើអេក្រង់

**cin >> student[i].age; //** បញ្ចូលតម្លៃឲ្យ age តាមរយៈ keyboard

**cout << "Sex : "; //** ការបង្ហាញអក្សរ Sex : មកលើអេក្រង់

**cin >> student[i].sex; //** បញ្ចូលតម្លៃឲ្យ sex តាមរយៈ keyboard

**cout << "Height : "; //** ការបង្ហាញអក្សរ Height : មកលើអេក្រង់

**cin >> student[i].height; //** បញ្ចូលតម្លៃឲ្យ height តាមរយៈ keyboard

**cout << "Weight : "; //** ការបង្ហាញអក្សរ Weight : មកលើអេក្រង់

**cin >> student[i].weight; //** បញ្ចូលតម្លៃឲ្យ weight តាមរយៈ keyboard

**}**

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << "Storing on the file ... \n";**

**for (i = 0; i < n; i++){**

**myfile.write((char\*) &student[i],**

**sizeof(student[i]));**

**}**

**myfile.close(); //** ការបិទ myfile

**// reading from the file**

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ...\n";**

**i = 0; //** ការកំណត់តម្លៃ 0 ទៅឲ្យអញ្ញាត i

**while(!myfile.eof()){ //** នៅពេល myfile មិនទាន់ដល់ទីបញ្ចប់ទេ នោះ

**myfile.read((char\*) &student[i],**

**​ sizeof(student[i]));**

**}**

**for (int j = 0; j < n; j++){**

**cout << student[j].name<<​ setw(5)<< student[j].age**

**<< setw(10) << student[j].sex << setw(5) <<**

**student[j].height << setw(5) <<**

**​ student[j].weight << endl;**

**}**

**myfile.close(); //** ការបិទ myfile

**} // end of main**

៧. ការប្រើ Array នៃ class ជាមួយនឹង File

នៅក្នុងជំពូក "Class និង Objects" យើងបានសិក្សារួចហើយនូវរបៀបបង្កើត និងប្រកាស array នៃ class object ក្នុងភាសា C++។ នៅក្នុងផ្នែកនេះ យើងនឹងសិក្សាពីរបៀបអាន និងសរសេរ class object តាមរយៈ file ដែលបានកំណត់។ យើងបានស្គាល់ហើយថា array មួយគឺជាប្រភេទទិន្នន័យដែលកំណត់ដោយអ្នកប្រើប្រាស់ ហើយធាតុរបស់វាមានលក្ខណៈ ដូចៗគ្នានិងផ្ទុកនៅក្នុងទីតាំង memory ជាប់ៗគ្នា។ ការអនុវត្តជាក់ស្តែង array នៃ class object គឺមានសារៈសំខាន់សម្រាប់ប្រព័ន្ធគ្រប់គ្រងទិន្នន័យសាំញ៉ាំ ហេតុនេះហើយវាមានប្រយោជន៍សម្រាប់សិក្សានូវរបៀបដែល array នៃ class objects ត្រូវអាន និងសរសេរដាក់ទៅក្នុង file មួយ។

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញនូវរបៀបអាន និងសរសេរ array នៃ class objects ពី file មួយ។

**#include <fstream.h>**

**int const max = 200;**

**class Student {**

**protected :**

**char name[20]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 20តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int មានលក្ខណៈ protected

**public :**

**void getdata(); //** ការប្រកាសអនុគមន៍ getdata() មានលក្ខណៈ public

**void display(); //** ការប្រកាសអនុគមន៍ display() មានលក្ខណៈ public

**};**

**void main(){**

**Student obj[max]; //** ការបង្កើត object student នៃ Student ចំនួន max

**fstream myfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ myfile

**myfile.open("data", ios::in || ios::out);**

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**.....**

**.....**

**cout << "Storing on the file ... \n";**

**for (i = 0; i < n; i++){**

**myfile.write((char\*) &obj[i], sizeof(obj[i]));**

**}**

**.....**

**.....**

**// reading from the file**

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ...\n";**

**for (i = 0; i < n; i++){**

**myfile.read((char\*) &obj[i], sizeof(obj[i]));**

**obj[i].display();//** ការហៅអនុគមន៍ display() មកប្រើតាមរយៈ obj នៃ Student class

**}**

**myfile.close(); //** ការបិទ myfile

**}**

ឧទាហរណ៍ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការអាន array នៃ class object របស់ Student ដូចជា name, age, sex, height និង weight ពី keyboard ហើយផ្ទុកវានៅលើ file មួយដែលបានកំណត់ដោយប្រើ member function read() និង write()។ បន្ទាប់មក file ដដែលនេះត្រូវបានបើកដើម្បីអាន និងបង្ហាញខ្លឹមសារនៃ file នៅលើអេក្រង់។

**// array of class objects and file operations**

**#include <fstream.h>**

**#include <iostream.h>**

**#include <iomanip.h>**

**int const max = 10;**

**class Student {**

**protected :**

**char name[10]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int មានលក្ខណៈ protected

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char មានលក្ខណៈ protected

**float height; //** ការប្រកាសអញ្ញាត height ជាប្រភេទ float មានលក្ខណៈ protected

**float weight; //** ការប្រកាសអញ្ញាត weight ជាប្រភេទ float មានលក្ខណៈ protected

**public :**

**void getdata(); //** ការប្រកាសអនុគមន៍ getdata() មានលក្ខណៈ public

**void display(); //** ការប្រកាសអនុគមន៍ display() មានលក្ខណៈ public

**};**

**void Student::getdata(){**

**cout << "Name : "; //** ការបង្ហាញអក្សរ Name : មកលើអេក្រង់

**cin >> name; //** បញ្ចូលតម្លៃឲ្យ name តាមរយៈ keyboard

**cout << "Age : "; //** ការបង្ហាញអក្សរ Age : មកលើអេក្រង់

**cin >> age; //** បញ្ចូលតម្លៃឲ្យ age តាមរយៈ keyboard

**cout << "Sex : "; //** ការបង្ហាញអក្សរ Sex : មកលើអេក្រង់

**cin >> sex; //** បញ្ចូលតម្លៃឲ្យ sex តាមរយៈ keyboard

**cout << "Height : "; //** ការបង្ហាញអក្សរ Height : មកលើអេក្រង់

**cin >> height; //** បញ្ចូលតម្លៃឲ្យ height តាមរយៈ keyboard

**cout << "Weight : "; //** ការបង្ហាញអក្សរ Weight : មកលើអេក្រង់

**cin >> weight; //** បញ្ចូលតម្លៃឲ្យ weight តាមរយៈ keyboard

**}**

**void Student::display(){**

**cout << name << setw(5) << age << setw(10) << sex**

**<< setw(5) << height << setw(5) << weight << endl;**

**}**

**void main(){**

**Student obj[max]; //** ការបង្កើត object ឈ្មោះ obj នៃ Student class ចំនួន max

**fstream myfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ myfile

**char fname[10]; //** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char ចំនួន 10តួអក្សរ

**cout << "Enter a file name to be stored ? \n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**myfile.open(fname, ios::in|ios::out);**

**cout << "How many objects are to be stored ? \n";**

**cin >> n; //** បញ្ចូលតម្លៃឲ្យ n តាមរយៈ keyboard

**// reading from the keyboard**

**cout << "Enter the following information \n";**

**for (i = 0; i < n; i++){**

**int j = i; //** ការកំណត់តម្លៃអញ្ញាត i ទៅឲ្យអញ្ញាត j

**cout << endl << "object = " << (j+1) << endl;**

**obj[i].getdata();//** ការហៅអនុគមន៍ display() មកប្រើតាមរយៈ obj នៃ Student class

**}**

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << "Storing on the file ... \n";**

**for (i = 0; i < n; i++){**

**myfile.write((char\*) &obj[i], sizeof(obj[i]));**

**}**

**myfile.close(); //** ការបិទ myfile

**// reading from the file**

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ...\n";**

**for (i = 0; i < n; i++){**

**myfile.read((char\*) &obj[i], sizeof(obj[i]));**

**obj[i].display();//**ការហៅអនុគមន៍ display() មកប្រើតាមរយៈ obj នៃ Student class

**}**

**myfile.close(); //** ការបិទ myfile

**}**

៨. ការប្រើ class មួយនៅក្នុង class មួយទៀតជាមួយនឹង File

នៅក្នុងជំពូកមុននេះស្តីអំពី "Classes និង Objects" វាបានលើកឡើងអំពី class មួយអាចជា member នៃ class មួយទៀត។ កាលណា class បានប្រកាសជា member នៃ class មួយទៀត នោះវាត្រូវបានហៅថា nested class ឬ class នៅក្នុង class មួយទៀត។ នៅពេល class មួយត្រូវបានប្រកាស ជា member នៃ class មួយទៀតនោះវាមានទំហំ scope ត្រឹមតែ class មួយទៀតប៉ុណ្ណោះ។ object នៃ class ខាងក្រៅពុំមានផ្ទុក object នៃ class ខាងក្នុងទេ។ នៅក្នុងផ្នែកនេះ ការអាននិងសរសេររបស់ nested class objects ពី file មួយត្រូវបានលើកឡើងដើម្បីសិក្សាលម្អិត។

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញនូវរបៀបអាននិងសរសេរ objects នៃ nested class ពី file មួយ។

**// array of nested class objects using file operations**

**#include <fstream.h>**

**class Student {**

**private :**

**char name[20]; //** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char ចំនួន 20តួអក្សរ

**public :**

**void getbase(); //** ការប្រកាសអនុគមន៍ getbase() មានលក្ខណៈ public

**void display(); //** ការប្រកាសអនុគមន៍ display() មានលក្ខណៈ public

**class Date {**

**private :**

**int year; //** ការប្រកាសអញ្ញាត year ជាប្រភេទ int មានលក្ខណៈ private

**public :**

**void getdate(); //** ការប្រកាសអនុគមន៍ getdate() មានលក្ខណៈ public

**void showDate();//** ការប្រកាសអនុគមន៍ showDate() មានលក្ខណៈ public

**class Age {**

**private :**

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int មានលក្ខណៈ private

**public :**

**void getAge();//** ការប្រកាសអនុគមន៍ getAge() មានលក្ខណៈ public

**void showAge(); //** ការប្រកាសអនុគមន៍ showAge() **}; // end of Age class**

**}; // end of Date class declaration**

**}; // end of Student class declaration**

**void main(){**

**Student obj1[5]; //** ការបង្កើត object ឈ្មោះ obj1 នៃ Student class ចំនួន 5

**Student::Date obj2[5];//** ការបង្កើត object ឈ្មោះ obj2 នៃ Date class ចំនួន 5

**Student::Date::Age obj3[5];**

**//** ការបង្កើត object ឈ្មោះ obj3 នៃ Age class ចំនួន 5

**fstream myfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ myfile

**myfile.open(fname, ios::in || ios::out);**

**.....**

**.....**

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << "Storing onto the file ... \n";**

**for (i = 0; i < n; i++){**

**//** ការសរសេរចូល file នូវតម្លៃផ្ទុកក្នុង object ទាំងបីប្រភេទ

**myfile.write((char\*) &obj1[i], sizeof(obj1[i]));**

**myfile.write((char\*) &obj2[i], sizeof(obj2[i]));**

**myfile.write((char\*) &obj3[i], sizeof(obj3[i]));**

**}**

**myfile.close(); //** ការបិទ myfile

**.....**

**.....**

**// reading from the file**

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ...\n";**

**for (i = 0; i < n; i++) {**

**//** ការទាញយកតម្លៃចេញពី file ទៅដាក់ក្នុង object ទាំងបីប្រភេទ

**myfile.read((char\*) &obj1, sizeof(obj1));**

**myfile.read((char\*) &obj2, sizeof(obj2));**

**myfile.read((char\*) &obj3, sizeof(obj3));**

**obj1.display();//** ការហៅអនុគមន៍ display() មកប្រើតាមរយៈ obj1 នៃ Student class

**obj2.showDate();//**ការហៅអនុគមន៍ showDate() មកប្រើតាមរយៈ obj2 នៃ Student class

**obj3.showAge();//**ការហៅអនុគមន៍ showAge() មកប្រើតាមរយៈ obj3 នៃ Student class

**}**

**myfile.close(); //** ការបិទ myfile

**}**

ឧទាហរណ៍ ៖ កម្មវិធីខាងក្រោមនេះបង្ហាញរបៀបអានទិន្នន័យពី array នៃ nested class objects តាមរយៈ keyboard ហើយសរសេរទិន្នន័យទៅដាក់ក្នុង file មួយដែលបានកំណត់។

**// array of nested class objects using file operations**

**#include <fstream.h>**

**#include <string.h>**

**const int max = 10;**

**class Student {**

**private :**

**char name[20];//** ការប្រកាសអញ្ញាត name ជា array ប្រភេទ char មានចំនួន 20តួអក្សរ

**long int id; //** ការប្រកាសអញ្ញាត id ជាប្រភេទ long int មានលក្ខណៈ private

**char sex; //** ការប្រកាសអញ្ញាត sex ជាប្រភេទ char មានលក្ខណៈ private

**public :**

**void getBase(); //** ការប្រកាសអនុគមន៍ getBase() មានលក្ខណៈ public

**void display(); //** ការប្រកាសអនុគមន៍ display() មានលក្ខណៈ public

**class Date {**

**private :**

**int day; //** ការប្រកាសអញ្ញាត day ជាប្រភេទ int មានលក្ខណៈ private

**int month; //** ការប្រកាសអញ្ញាត month ជាប្រភេទ int មានលក្ខណៈ private

**int year; //** ការប្រកាសអញ្ញាត year ជាប្រភេទ int មានលក្ខណៈ private

**public :**

**void getDate(); //** ការប្រកាសអនុគមន៍ getDate() មានលក្ខណៈ public

**void showDate();//** ការប្រកាសអនុគមន៍ showDate() មានលក្ខណៈ public

**class Age {**

**private :**

**int age; //** ការប្រកាសអញ្ញាត age ជាប្រភេទ int មានលក្ខណៈ private

**public :**

**void getAge();//**ការប្រកាសអនុគមន៍ getAge() មានលក្ខណៈ public

**void showAge(); //** ការប្រកាសអនុគមន៍ showAge()

**}; // end of age class**

**}; // end of date class declaration**

**}; // end of student class declaration**

**void Student::getBase(){**

**cout << "Enter a name :"; //** ការបង្ហាញអក្សរ Enter a name : មកលើអេក្រង់

**cin >> name; //** បញ្ចូលតម្លៃឲ្យ name តាមរយៈ keyboard

**cout << "ID :"; //** ការបង្ហាញអក្សរ ID : មកលើអេក្រង់

**cin >> id; //** បញ្ចូលតម្លៃឲ្យ id តាមរយៈ keyboard

**cout << "Sex :"; //** ការបង្ហាញអក្សរ Sex : មកលើអេក្រង់

**cin >> sex; //** បញ្ចូលតម្លៃឲ្យ sex តាមរយៈ keyboard

**}**

**void Student::date::getDate() {**

**cout << "Enter a date of birth " << endl;**

**cout << "Day :"; //** ការបង្ហាញអក្សរ Day : មកលើអេក្រង់

**cin >> day; //** បញ្ចូលតម្លៃឲ្យ day តាមរយៈ keyboard

**cout << "Month :"; //** ការបង្ហាញអក្សរ Month : មកលើអេក្រង់

**cin >> month; //** បញ្ចូលតម្លៃឲ្យ month តាមរយៈ keyboard

**cout << "Year :"; //** ការបង្ហាញអក្សរ Year : មកលើអេក្រង់

**cin >> year; //** បញ្ចូលតម្លៃឲ្យ year តាមរយៈ keyboard

**}**

**void Student::date::Age::getAge(){**

**cout << "Enter an age :"; //** ការបង្ហាញអក្សរ Enter an age : មកលើអេក្រង់

**cin >> age; //** បញ្ចូលតម្លៃឲ្យ age តាមរយៈ keyboard

**}**

**void Student::display(){**

**cout << name << " " << '\t'; //** បង្ហាញតម្លៃ name មកលើអេក្រង់

**cout << id << " " ; //** បង្ហាញតម្លៃ id មកលើអេក្រង់

**cout << sex << " "; //** បង្ហាញតម្លៃ sex មកលើអេក្រង់

**}**

**void Student::date::showDate(){**

**cout << day << '/' << month << '/' << year << '\t';**

**}**

**void Student::date::Age::showAge(){**

**cout << '\t' << age << endl;**

**}**

**void main(){**

**Student obj1[max]; //** ការបង្កើត object ឈ្មោះ obj1 នៃ Student class ចំនួន max

**Student::Date obj2[max];//**ការបង្កើត object ឈ្មោះ obj2 នៃ Date class ចំនួន max

**Student::Date::Age obj3[max];**

**//** ការបង្កើត object ឈ្មោះ obj3 នៃ Age class ចំនួន max

**int n, i; //** ការប្រកាសអញ្ញាត n, i ជាប្រភេទ int

**fstream myfile; //** ការបង្កើត object នៃ fstream ឈ្មោះ myfile

**char fname[10];//** ការប្រកាសអញ្ញាត fname ជា array ប្រភេទ char មានចំនួន 10តួអក្សរ

**cout << "Enter a file name to be stored :\n";**

**cin >> fname; //** បញ្ចូលតម្លៃឲ្យ fname តាមរយៈ keyboard

**myfile.open(fname,ios::in|ios::out);//**ការបើក file មួយតាមរយៈ fname

**cout << "How many students?\n";**

**cin >> n; //** បញ្ចូលតម្លៃឲ្យ n តាមរយៈ keyboard

**// reading from the keyboard**

**cout << "Enter the following information :\n";**

**for (i = 0; i < n; i++) {**

**int j = i + 1;**

**cout << "\n object : " << j << endl;**

**obj1[i].getBase(); //** ការហៅអនុគមន៍ getBase() មកប្រើតាមរយៈ obj1

**obj2[i].getDate(); //** ការហៅអនុគមន៍ getDate() មកប្រើតាមរយៈ obj2

**obj3[i].getAge(); //** ការហៅអនុគមន៍ getAge() មកប្រើតាមរយៈ obj3

**}**

**// storing onto the file**

**myfile.open(fname, ios::out); //** ការបើក file មួយតាមរយៈ fname

**cout << "Storing onto the file ... \n";**

**for (i = 0; i < n; i++){**

**//** ការសរសេរចូល file នូវតម្លៃផ្ទុកក្នុង object ទាំងបីប្រភេទ

**myfile.write((char\*) &obj1[i], sizeof(obj1[i]));**

**myfile.write((char\*) &obj2[i], sizeof(obj2[i]));**

**myfile.write((char\*) &obj3[i], sizeof(obj3[i]));**

**}**

**myfile.close(); //** ការបិទ myfile

**// reading from the file**

**myfile.open(fname, ios::in); //** ការបើក file មួយតាមរយៈ fname

**cout << "Reading from the file ...\n";**

**cout << "\n\n\n" << endl;**

**cout << "Contents of the array of nested classes\n";**

**cout << "-------------------------------------"<<endl;**

**cout << "Student's name ID\_no Sex DateOfBirth Age\n";**

**cout << "-------------------------------------"<<endl;**

**for (i = 0; i < n; i++) {**

**//** ការទាញយកតម្លៃចេញពី file ទៅដាក់ក្នុង object ទាំងបីប្រភេទ

**myfile.read((char\*) &obj1[i], sizeof(obj1[i]));**

**myfile.read((char\*) &obj2[i], sizeof(obj2[i]));**

**myfile.read((char\*) &obj3[i], sizeof(obj3[i]));**

**obj1[i].display(); //** ការហៅអនុគមន៍ display() មកប្រើតាមរយៈ obj1

**obj2[i].showDate(); //** ការហៅអនុគមន៍ showDate() មកប្រើតាមរយៈ obj2

**obj3[i].showAge(); //** ការហៅអនុគមន៍ showAge() មកប្រើតាមរយៈ obj3

**}**

**cout << "------------------------------------"<<endl;**

**myfile.close(); //** ការបិទ myfile

**}**

៩. ដំណើរការ file ដោយចូលេប្រី data មិនតាមលំដាប់លំដោយ

រហូតមកដល់ពេលនេះ file ដែលមានលក្ខណៈតាមលំដាប់លំដោយអាចប្រកាស និងចូលប្រើយ៉ាងណានោះនៅក្នុង C++ បានត្រូវបានពន្យល់រួចហើយ។ ការចូលប្រើ file ដែលមានលក្ខណៈលំដាប់លំដោយនោះមានភាពងាយស្រួលបង្កើតជាង random access file។ ចំពោះ file ដែលមានលក្ខណៈលំដាប់លំដោយនេះ ទិន្នន័យត្រូវបានផ្ទុកនិងទាញយកមកប្រើវិញបន្តបន្ទាប់គ្នាពីមួយទៅមួយទៀត។ file pointer ជានិច្ចជាកាលផ្លាស់ទីពីដើម file ទៅចុង file។ ផ្ទុយទៅវិញ random access file មិនត្រូវការចាប់ផ្តើមពីដើម file ទៅចុង file ឡើយ។ នេះមានន័យថា ការចូលប្រើមិនតាមលំដាប់លំដោយអាចផ្លាស់ទី file pointer ដោយផ្ទាល់ទៅកាន់ ទីតាំងណាមួយនៅក្នុង file ជំនួសឲ្យការផ្លាស់ទី file pointer តាមលំដាប់លំដោយ។ វិធីបែបនេះត្រូវបានគេប្រើនៅក្នុង database files។ ដើម្បីធ្វើការអាន និងកែតម្រូវ object នៃ database នោះ file គួរតែប្រើ mode ប្រភេទ៖ អាន និង សរសេរ។ header file ឈ្មោះ <fstream.h> ត្រូវការប្រកាសឲ្យ random access file។ ដូចបានពោលនៅក្នុងផ្នែកមុននេះដែរថា fstream គឺជា base class របស់ ifstream និង ofstream។ fstream បានទទួលនូវ file pointers ពីរ គឺមួយសម្រាប់ input buffer និងមួយទៀតសម្រាប់ output buffer ដើម្បីប្រើ random access file ក្នុងការអាននិងសរសេរ។

**** ការប្រកាស random access file**៖** random access file ត្រូវបានបើកដោយប្រើ mode នៃ ការចូលប្រើដូចខាងក្រោម ៖

|  |  |
| --- | --- |
| Mode **នៃការចូលប្រើ** | **អត្ថន័យ** |
| ios::in  ios::out  ios::ate  ios::binary | សម្រាប់អាន file មួយ  សម្រាប់សរសេរ file មួយ  សម្រាប់សរសេរនិងអាន (random)  កំណត់យកទម្រង់ជា binary |

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញពីរបៀប random access file ត្រូវបានបើកសម្រាប់អាន និងសរសេរ។

**#include <fstream.h>**

**void main(){**

**fstream file;**

**file.open(fname, ios::in |ios::out |ios::ate |ios::binary);**

**.....**

**.....**

**}**

ចំពោះការបើក random access file ជាមួយនឹង mode នៃការចូលប្រើលក្ខណៈក្នុងតារាងខាងក្រោមនេះ វាមានសារៈសំខាន់ណាស់សម្រាប់អាន, សរសេរ និងសរសេរបន្ថែម។ file ត្រូវតែប្រកាសជាលក្ខណៈ binary ដោយសារ data member នៃ class មួយត្រូវផ្ទុកក្នុងទម្រង់ជា binary។

fstream ទទួលលក្ខណៈ member functions ខាងក្រោមនេះដើម្បីផ្លាស់ទី file pointer នៅខាងក្នុងនិងជុំវិញ database។

|  |  |
| --- | --- |
| តម្លៃដែលមាន | ទីតាំង file |
| ios::beg  ios::cur  ios::end | ពីដើមដំបូងនៃ file  ពីទីតាំង file pointer កំពុងប្រើ  ពីទីបញ្ចប់នៃ file |

Member functions ខាងក្រោមនេះត្រូវបានប្រើ សម្រាប់ដំណើរការ random access file។

ក. seekg( )៖ member function ឈ្មោះ seekg() ត្រូវបានប្រើសម្រាប់រកទីតាំងនៃការប្រតិបត្តិ file ចំពោះការប្រតិបត្តិបញ្ចូលទិន្នន័យមិនតាមលំដាប់លំដោយ។

ឧទាហរណ៍ code ខាងក្រោមនេះបង្ហាញនូវការរកទីតាំងនៃការប្រតិបត្តិ file ចំពោះ random access file។

**#include <fstream.h>**

**void main(){**

**fstream myfile; //** ការប្រកាស object នៃ fstream ឈ្មោះ myfile

**.....**

**.....**

**myfile.seekg(40); //** នាំទៅកាន់ byte លេខទី 40

**myfile.seekg(40, ios::beg); //** មានលក្ខណៈដូចឃ្លាខាងលើ

**myfile.seekg(0, ios::end); //** នាំទៅកាន់ផ្នែកខាងចុងនៃ file

**myfile.seekg(0); //** នាំទៅកាន់ផ្នែកខាងដើមនៃ file

**myfile.seekg(-1, ios::cur); //** file pointer ត្រូវបានរំកិលថយក្រោយមួយ byte

**}**

ខ. seekp( ) ៖ member function ឈ្មោះ seekp() ត្រូវបានប្រើសម្រាប់រកទីតាំងនៃការប្រតិបត្តិ file ចំពោះការប្រតិបត្តិបញ្ចេញទិន្នន័យមិនតាមលំដាប់លំដោយ។

គ. tellg( )៖ member function ឈ្មោះ tellg() ត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើលទីតាំងកំពុងប្រើរបស់ input stream។

ឃ. tellp( )៖ member function ឈ្មោះ tellp() ត្រូវបានប្រើសម្រាប់ផ្ទៀងផ្ទាត់មើលទីតាំងកំពុងប្រើរបស់ output stream។

៩.១ អំពី File Pointers ជាមយួនឹងការប្រើរបស់វា

file និមួយៗមាន pointers ពីរដែលមានទំនាក់ទំនងគ្នាហៅថា file pointers។ file pointer មួយមានឈ្មោះថា input pointer (ឬ get pointer) ហើយមួយទៀតត្រូវបានហៅថា output pointer (ឬ put pointer)។ គេប្រើ pointers ទាំងនេះសម្រាប់ផ្លាស់ទីក្នុង files ពេលអានឬសរសេរ។ input pointer ត្រូវបានប្រើសម្រាប់អាន contents នៃ file ត្រង់ទីតាំងមួយនិង output pointer ត្រូវបានប្រើសម្រាប់សរសេរទៅកាន់ទីតាំងមួយនៃ file ដែលគេកំណត់។ នៅពេលចាប់ធ្វើការប្រតិបត្តិលើផ្នែក input ឬ output នោះ pointer ដែលត្រូវគ្នានឹងដំណើរការទៅមុខដោយស្វ័យប្រវត្ត។

៩.២ សកម្មភាពតាមធម្មតា (default actions)

នៅពេលគេបើក file មួយក្នុងលក្ខណៈ read-only mode នោះ input pointer ត្រូវបានដាក់នៅត្រង់ចំណុចចាប់ផ្តើមដោយស្វ័យប្រវត្ត ហេតុនេះគេអាចអាន file ពីចំណុចចាប់ផ្តើម បាន។ ដូចគ្នានេះដែរ នៅពេលគេបើក file មួយក្នុងលក្ខណៈ write-only mode វិញ contents ដែលមានស្រាប់ត្រូវបានលុប ហើយ output pointer ត្រូវបានដាក់នៅត្រង់ចំណុចចាប់ផ្តើម។ លក្ខណៈនេះអាចឲ្យគេសរសេរ file ពីចំណុចចាប់ផ្តើម។ នៅក្នុងករណីនេះ គេចង់បើក file ដែលមានស្រាប់ដើម្បីបន្ថែមទិន្នន័យទៀតនោះ គេត្រូវតែបើក file ក្នុងលក្ខណៈជា "append" mode។ លក្ខណៈនេះនឹងនាំ output pointer ទៅកាន់ផ្នែកខាងចុងនៃ file។

៩.៣ Functions សម្រាប់ប្រើ file pointers

គ្រប់សកម្មភាពលើ file pointers ទាំងអស់ប្រព្រឹត្តទៅតាមលក្ខណៈធម្មតាដោយស្វ័យប្រវត្ត។ តើគេអាចផ្លាស់ទី file pointer ទៅកាន់ទីតាំងណាមួយនៅក្នុង file តាមតម្រូវការយ៉ាងដូចម្តេច? គេអាចរៀបចំធ្វើការផ្លាស់ទី file pointer ដោយខ្លួនគេបាន ព្រោះវាមាន functions មួយចំនួនរបស់ file stream class ជួយបំពេញនូវតម្រូវការទាំងនេះ។

seekg() រំកិល get pointer (input) ទៅកាន់ទីតាំងកំណត់

seekp() រំកិល put pointer (output) ទៅកាន់ទីតាំងកំណត់

tellg() ឲ្យតម្លៃទីតាំងកំពុងឈររបស់ get pointer

tellp() ឲ្យតម្លៃទីតាំងកំពុងឈររបស់ put pointer

ឧទាហរណ៍ ៖ **infile.seekg(10);**

ឃ្លានេះមានន័យថាផ្លាស់ទី file pointer ទៅកាន់ byte លេខ 10។ ចូរកត់សម្គាល់ថា bytes នៅ ក្នុង file មួយត្រូវបានដាក់លេខចាប់ពីសូន្យ។ ហេតុនេះ pointer នឹងស្ថិតនៅត្រង់ byte ទី 11 ក្នុង file។

ចូរសង្កេតឃ្លាខាងក្រោមនេះ ៖

**ofstream fout;**

**fout.open("hello", ios::app);**

**int p = fout.tellp();**

តាមការប្រតិបត្តិនៃឃ្លាទាំងនេះ output pointer ត្រូវបានផ្លាស់ទីទៅដល់ផ្នែកខាងចុងនៃ file "hello" ហើយតម្លៃ p នឹងតាងឲ្យចំនួន bytes របស់ file នោះ។

៩.៤ ការកំណត់ offset

យើងទើបតែឃើញនូវរបៀបផ្លាស់ទី file pointer មួយទៅកាន់ទីតាំងតម្រូវការដោយប្រើ "seek" functions។ arguments សម្រាប់ functions ទាំងនេះតាងឲ្យទីតាំងជាក់លាក់នៅក្នុង file មួយ។

**ឧទាហរណ៍ ៖** outfile.seekp(m);

**END**

**START**

**FILE**

m bytes file pointer

ទម្រង់ "seek" functions មានដូចខាងក្រោម ៖

seekg(offset, refposition);

seekp(offset, refposition);

ប៉ារ៉ាម៉ែត្រ offset តាងឲ្យចំនួន bytes ដែល file pointer ត្រូវផ្លាស់ទីពីតាំងកំណត់ដោយ ប៉ារ៉ាម៉ែត្រ refposition។ refposition មានតម្លៃថេរកំណត់ដោយ ios class ជា ៖

ios::beg (=0) ចាប់ផ្តើមពី file

ios::cur (=1) ទីតាំងកំពុងប្រើរបស់ pointer

ios::end (=2) ខាងចុង file

ឧទាហរណ៍ ៖

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 0

in.seekg(4, ios::beg); ឬ in.seekg(4);

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 4

in.seekg(1, ios::cur);

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 5

in.seekg(–3 , ios::cur);

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 2

in.seekg(0, ios::beg);

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 0

in.seekg(0, ios::end);

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 10

in.seekg(–2 , ios::end);

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

file pointer = 8

កម្មវិធីខាងក្រោមនេះបង្ហាញពីការប្រើ random access file។

**#include <iostream.h>**

**#include <fstream.h>**

**void main(){**

**char c; //** ការប្រកាសអញ្ញាត c ជាប្រភេទ char

**long int file\_pos, last\_pos, offset;**

**//** ការប្រកាសអញ្ញាត file\_pos, last\_pos, offset ជាប្រភេទ long int

**ifstream in("letters.txt"); //** ការបង្កើត object នៃ ifstream ឈ្មោះ in

**if (in.fail()){**

**cout << "Error: failed in opening file.\n";**

**return;**

**}**

**//** រកទីតាំងរបស់តួអក្សរនៅខាងចុងបំផុតក្នុង file

**in.seekg(0, ios::end);**

**last\_pos = in.tellg();//**ការហៅអនុគមន៍ tellg() តាមរយៈ object នៃ ifstream ឈ្មោះ in

**//** ការប្រតិបត្តិដដែលៗរហូតដល់អ្នកប្រើប្រាស់បញ្ចូលតម្លៃ 0

**do {**

**cout << "Enter the file position(0 to end)";**

**cin >> file\_pos; //** បញ្ចូលតម្លៃឲ្យ file\_pos តាមរយៈ keyboard

**//** ផ្ទៀងផ្ទាត់ថាតើវាមានទីតាំងត្រឹមត្រូវឬទេ?

**if (file\_pos > last\_pos || file\_pos < 0)**

**cout << "Invalid position. Enter a value "**

**<< "between 1 and " << last\_pos << endl;**

**if (file\_pos > 0 && file\_pos <= last\_pos) {**

**// The offset is 1 less than the position.**

**offset = file\_pos-1;**

**//** ផ្លាស់ទីទៅកាន់ទីតាំង offset នៃតួអក្សរ

**in.seekg(offset, ios::beg);**

**//** បញ្ចូលតម្លៃជាតួអក្សរ និង បង្ហាញតម្លៃមកលើអេក្រង់

**in.get(c);**

**cout << "Character at position " << file\_pos**

**<< " , offset " << offset << " is " << c**

**<< " ASCII " << (int)(c) << endl;**

**}**

**} while(file\_pos != 0);**

**}**

សំណួរនិងលំហាត់

១ - តើ file ចែកជាប៉ុន្មានប្រភេទ? គឺអ្វីខ្លះ? ចូរនិយាយពីសារៈសំខាន់នៃ file។

២ - តើ file stream ដែលប្អូនបានសិក្សាមានអ្វីខ្លះ? ចូរប្រាប់ពី header file របស់វាផង។

៣ - ចូរសរសេរ code បង្ហាញពីការបង្កើត file stream ដើមី្ប write file ដោយបញ្ចូលអក្សរពី

keyboard និង read file នេះមកបង្ហាញមកលើអេក្រង់។

៤ - ចូរសរសេរ code បង្ហាញពីការ​ read text file មួយរួចបង្ហាញមកលើអេក្រង់ ជាអក្សរធំ

(capital letter)។

៥ - ចូរសរសេរ code បង្ហាញពីការបង្កើត file stream ដើមី្ប write file ជាទម្រង់ binary ដោយ

​​​​ បញ្ចូលតម្លៃពី keyboard និង read តម្លៃពី file នោះមកបង្ហាញមកលើអេក្រង់។

៦- គេមាន class មួយដូចខាងក្រោមៈ ​​

**class Emp {**

**protected :**

**int id, age;**

**char name[20];**

**float salary, rate, hour;**

**public :**

**Emp(int i, char na[], float sal, float r, float hr);​**

**//** ជា **constructor**

**void display(); //** បង្ហាញព័ត៌មានមកលើអេក្រង់

**float income(); //** ប្រាក់ចំណូល=ប្រាក់ខែ+(កំរៃប្រាក់ម៉ោង X ចំនួនម៉ោងធ្វើការលើស)

**int operator>(Emp obj); //** ប្រៀបធៀបប្រាក់ខែបុគ្គលិកនៃ Emp obj ចំនួន២

**};**

ក - ចូរសរសេរ function definitions ទាំងអស់របស់ class ខាងលើ។

ខ- ចូរបង្កើត object នៃ class ខាងលើឲ្យបានចំនួន ៣ objects ដោយកំណត់តម្លៃដំបូងទៅ

ឲ្យវា។

គ- ចូរសរសេរ code ដើមី្ប write objects ទាំង៣ខាងលើ ទៅដាក់ក្នុង file មួយឈ្មោះ

mydata.bin។

ឃ- ចូរសរសេរ code ដើមី្ប read objects ទាំង៣ខាងលើ ចេញពី file ឈ្មោះ mydata.bin។

ឯកសារយោង៖

* Programming with C++, B.L. Juneja and Anita Seth, Copyright © 2009, New Age International (P) Ltd. , New Delhi, India.
* Sams Teach Yourself C++ in 24 Hours, Jesse Liberty and Rogers Cadenhead, Copyright © 2011 by Pearson Education, Inc., Printed in the United States of America.