

**الوحدة الثانية عشرة**

**معالجة الملفات File Processing**

12.0

## بنهاية هذه الوحدة:

1. ستتمكن من التعامل مع الدفق وتتعرف على الملفات التتابعية.
2. ستتمكن من إنشاء ملفات تتابعية، والتعامل معها.
3. ستتمكن من الوصول إلى السجلات المختلفة تتابعياً.
4. ستتمكن من الوصول إلى السجلات المختلفة عشوائياً.

**مقدمة**

**12.1**

تخزين البيانات في المتغيرات أو المصفوفات هو تخزين مؤقت، لذلك نحتاج الى وسيلة تخزين دائمة. وتوفر الملفات Files هذه الوسيلة.

يخزن الحاسوب الملفات في وسائط التخزين الثانوية مثل الأقراص.

في هذه الوحدة، سنوضح كيفية إنشاء ومعالجة الملفات من خلال برامج لغة C++.

عادة تتكون المفات من مجموعة من السجلات Records والتي تتكون بدورها من مجموعة من الحقول Fields. يتكون ملف للموظفين مثلاً على مجموعة من السجلات (سجل لكل موظف)، وقد يحتوي السجل مثلاً على الحقول التالية:

1. رقم الموظف.
2. إسم الموظف.
3. العنوان.
4. المرتب.

لتسهيل الوصول الى سجل ما في ملف، يتم اختيار حقل مفتاحي للسجل Record Key. والذي يجب أن يكون فريداً Unique في الملف.

في ملف الموظفين اعلاه، يمكن اختيار رقم الموظف كحقل مفتاحي للملف.

هناك عدة طرق لتنظيم السجلات داخل الملف، أشهر الطرق المستخدمة هي الملفات التتابعية Sequential Files والتي يتم فيها تخزين السجلات بترتيب حقولها المفتاحية، فمثلاً في ملف الموظفين، يكون أول سجل هو السجل الذي يحمل أقل رقم موظف.

### الملفات والدفق

#### Files and Stream

**12.2**

تعامل C++ الملفات كفيض متتابع من الثمانيات Bytes. الشكل التالي يوضح ملف يتكون من n Byte

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**n-1**

**end**

**………………**

عند فتح ملف يتم إنشاء كائن يقترن معه الدفق. لقد رأينا من قبل أربعة كائنات منشأة أتوماتيكياً، وهي cout ، cin ، cerr و clog.

يستخدم الكائن cin لإدخال بيانات من لوحة المفاتيح، والكائن cout يستخدم لإخراج بيانات إلى الشاشة، والكائنان cerr و clog يستخدمان لإخراج رسائل الأخطاء إلى الشاشة.

عند التعامل مع الملفات، يجب تضمين ملفي الترويسةiostream.h و fstream.h حيث يحتوي الملف fstream.h على فئات الدفق ifstream ( والتي تستخدم في إدخال بيانات إلى الملفات) و ofstream (والتي تستخدم لإخراج بيات من المفات)، و fstream (لإدخال وإخراج بيانات من الملفات).

لفتح ملف، نحتاج لإنشاء كائن يتبع لإحدى هذه الفئات.

### إنشاء ملف تتابعي

#### **Creating a Sequential file**

**12.3**

لا تتطلب C++ أي هيكلية معينة للملف، وعليه لا يوجد مصطلح سجلات في ملفات C++ لذا يجب على المبرمج تحديد الكيفية التي يتم بها تنظيم الملف.

البرنامج التالي يوضح كيفية إنشاء ملف تتابعي:

//Program 12-1

//Creating a sequential file

**#include**<iostream.h>

**#include**<fstream.h>

**#include**<stdlib.h>

**main**( )

{

**ofstream** outclientfile(“clients.dat”,ios::out);

**if** (!outclientfile){

cerr<<”File could not be opened”<<endl;

exit (1);

}

**cout**<<”Enter the account, name, and balance.”

<<endl

<<”(Enter EOF to end input)”<<endl

<<”? “;

**int** account;

**char** name[10];

//Continued

**float** balance;

**while**(cin>>account>>name>>balance){

outclientfile<<account<<” “<<name<<” “<<balance

<<endl;

**cout**<<”? “;

}

**return** 0;

}

الخرج من البرنامج:

|  |
| --- |
| Enter the account, name, and balance.  (Enter EOF to end input)  ? **100 Ahmed 24.98**  ? **200 Ali 345.67**  ? **300 Hassan 0.00**  ? **400 Omer –42.16**  ? **500 Abbas 224.62**  ? **^Z** |

البرنامج السابق ينشئ ملفاً تتابعياً، حيث يمكن استخدامه في نظام حسابات مثلاً ليساعد في إدارة حسابات العملاء.

لكل عميل من العملاء، يتحصل البرنامج على رقم حساب العميل account وإسم العميل name ورصيد العميل balance. البيانات التي يتحصل عليها البرنامج لكل عميل تمثل سجل ذلك العميل.

يستخدم رقم حساب العميل كحقل مفتاحي، وعليه يكون الملف مرتباً بترتيب أرقام حسابات العملاء.

تم فتح الملف للكتابة فيه، لذلك ينشئ البرنامج كائن خرج تابع للفئة ofstream يدعى outclientfile، وتم تمرير وسيطتين لمشيد ذلك الكائن وهما إسم الملف Clients.dat، طريقة فتح الملف (File open mode) ios::out

يقوم البرنامج، باستقبال البيانات المدخلة وحفظها في الملف، إلى أن يتم إدخال رمز نهاية الملف (<**ctrl**> Z).

خرج البرنامج يفترض أنه تم إدخال بيانات خمسة عملاء، ثم تم إدخال رمز نهاية الملف **^Z.**

نلاحظ أننا قمنا بتضمين ملف الترويسة stdlib.h الذي يحتوي على تعريف الدالة exit، والتي تنهي البرنامج في حالة عدم فتح الملف بصورة صحيحة.

#### قراءة البيانات من ملف تتابعي

##### Reading Data from a Sequential file

**12.4**

سنقوم الآن بكتابة برنامج يقوم بقراءة الملف السابق، وطباعة محتوياته على الشاشة:

Program 12-2:

//Reading and printing a Sequential file

**#include<**iostream.h>

**#include**<fstream.h>

**#include**<iomanip.h>

**#include**<stdlib.h>

**void** outputline(int, char \*, float);

**main( )**

{

ifstream inClientFile(“clients.dat”,ios::in);

**if** (!inClientFile) {

**cerr** << “File could not be opened” <<endl;

exit(1);

}

**int** account;

**char** name[10];

//Continued

**float** balance;

**cout** <<setiosflags(ios::left) <<setw(10) <<”Account”

<<setw(13) <<”Name” <<”Balance”<<endl;

**while**(inClientFile >> account >.name >>balance)

outputline(account, name, balance);

return 0;

}

**void** outputline(int acct, char \*name, float bal)

{

**cout** << setiosflags(ios::left) << setw(10)<< acct

<< setw(13) << name<< setw(7)

<< setprecision(2)

<< setiosflags(ios::showpoint | ios::right)

<< bal << endl;

}

الخرج من البرنامج:

|  |
| --- |
| Account Name Balance  100 Ahmed 24.98  200 Ali 345.67  300 Hassan 0.00  400 Omer -42.16  500 Abbas 224.62 |

يتم فتح الملفات لقراءة بيانات منها بإنشاء كائن يتبع للفئة ifstream والذي يتم تمرير وسيطتين له هما إسم الملف clients.dat وطريقة فتح الملف File Open mode. فالإعلان:

ifstream inClientFile(“clients.dat”,ios::in);

ينشئ كائن تابع للفئة ifstream يدعى inClientFile، ليقوم بفتح الملف

clients.dat للقراءة منه.

###### الوصول العشوائي لمحتويات ملف تتابعي

**Random Access to a Sequential file**

**12.56**

يملك كل كائن ملف، مؤشرين مقترنين به يسميان مؤشر الحصول get pointer ومؤشر الوضع put pointer، ويسميان أيضاً مؤشر الحصول الحالي ومؤشر الوضع الحالي.

في بعض الأحيان، قد نرغب في بدء قراءة الملف من بدايته ومتابعته إلى نهايته، وقد ترغب عند الكتابة البدء من البداية وحذف أي محتويات موجودة ،لكن هنالك أوقات نحتاج فيها إلى التحكم بمؤشرات الملفات. لكي نتمكن من القراءة أو الكتابة في مواقع عشوائية من الملف.

تتيح الدالتان seekg و seekp ضبط مؤشري الحصول والوضع على التوالي.

يمكن استخدام الدوال seekg( ) و seekp( ) بطريقتين :-

1/ مع وسيطة واحدة هي موقع البايت المطلق في الملف ( بداية الملف هي البايت 0 ).

2/ مع وسيطتين الأولى إزاحة من موقع معين في الملف والثانية الموقع الذي تم قياس الإزاحة منه.

هنالك ثلاثة احتمالات للوسيطة الثانية:-

1. beg وهى بداية الملف.
2. Cur وتعنى الموقع الحالي للمؤشر.
3. End وتعنى نهاية الملف.

فمثلاً العبارة :-

seekp( -10,ios:: end);

ستضع مؤشر الوضع 10 بايتات قبل نهاية الملف.

البرنامج التالي يستخدم الدالة seekg مع وسيطة واحدة:

Program 12-3:

**#include**<iostream.h>

**#include**<fstream.h>

**#include**<iomanip.h>

**#include**<stdlib.h>

**void** outputline(**int**, **char** \*, **float**);

**main( )**

{

//Continued

ifstream inClientFile(“clients.dat”, ios::in);

**if** (!inClientFile){

**cerr**<< “File could not be opened” <<**endl**;

**exit**(1);

}

**cout** <<”Enter request “<<**endl**

<<” 1 – List accounts with zero balances” << **endl**

<<” 2 – List accounts with credit balances”<<**endl**

<<” 3 – List accounts with debit balances”<<**endl**

<<” 4 – End of run” <<**endl**<<”?”;

**int** request;

//Continued

**cin**>> request;

**while**(request !=4) {

**int** account;

**char** name[10];

**float** balance;

inClientFile >>account >>nam>>balance;

**switch** (request) {

**case** 1:

**cout** <<endl <<”Account with zero balances:”

<<**endl**;

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

**if** (balance==0)

outputline(account, name, balance);

inClientFile >>account >>name >>balance;

}

**break**;

**case** 2:

**cout**<<**endl**<<”Accounts with credit balance:”

<<**endl**;

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

**if** (balance <0)

outputline(account, name, balance);

//Continued

inClientFile>>account >>name >>balance;

}

**break**;

**case** 3:

**cout**<<**endl**<<”Accounts with debit balances:”

<<**endl**;

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

**if** (balance > 0)

outputline(account, name, balance);

inClientFile >>account>>name>>balance;

}

**break**;

}

inClientFile.clear( ); //reset eof for next input

inClientfile.seekg(0); //position to beginning of file

**cout**<<**endl** <<”? “;

**cin**>>request;

}

**cout** << “End of run.” <<**endl**;

**return** 0;

}

**cout** << setiosflags(ioa::left) << setw(10) << acct

<< setw(13) << name <<setw(7) <<setprecision(2)

<< setiosflags(ios::showpoint | ios::right)

<<bal <<endl;

}

الخرج من البرنامج:

|  |
| --- |
| Enter request  1 – List accounts with zero balances  2 – List accounts with credit balances  3 – List accounts with debit balances  4 – End of run  ?**1**  Accounts with zero balances:  300 Hassan 0.00  ?**2**  Accounts with credit balances:  400 Omer -42.16  ?**3**  Accounts with debit balances:  100 Ahmed 24.98  200 Ali 345.67  500 Abbas 224.62    ?**4**  End of run. |



**الملخص:**

* الملفات هي وسيلة دائمة لتخزين البيانات.
* تتكون الملفات عادة من مجموعة من السجلات.
* تتكون السجلات من مجموعة من الحقول.
* يكون لكل سجل حقل مفتاحي.
* في الملفات التتابعية يتم تخزين السجلات بترتيب حقولها المفتاحية.
* عند التعامل مع الملفات يجب تضمين الملف fstream.h.
* عند فتح ملف للكتابة فيه يجب إنشاء كائن تابع للفئة ofstream.
* يتم فتح الملفات لقراءة بيانات منها بإنشاء كائن يتبع الفئة ifstream.
* لإسترجاع بيانات من ملف تتم قراءة الملف من بدايته وقراءة كل محتويات الملف بالتتابع حتى نصل إلى البيانات المطلوبة.
* يملك كل كائن ملف مؤشرين مقترنين به يسميان مؤشر الحصول get pointer ومؤشر الوضع Put pointer.
* تضبط الدالتان seekg( ) و seekp( ) مؤشري الحصول والوضع على التوالي.

##### الأسئلة



1. **أنشئ ملف للموظفين يدعى Employee على أن يحتوي كل سجل في الملف على الحقول التالية:-**

* **رقم الموظف.**
* **إسم الموظف.**
* **العنوان.**

**ثم قم بإدخال بيانات خمسة موظفين.**

1. **تأكد من إدخال البيانات في السؤال السابق بصورة صحيحة وذلك بكتابة برنامج لقراءة محتويات الملف.**
2. **قم بكتابة برنامج يقوم باستقبال معلومات عن طلاب كلية ويضعها في ملف يسمى Students، بحيث يحتوي ملف الطلاب على الآتي:**

* **رقم الطالب.**
* **إسم الطالب.**
* **تخصص الطالب.**
* **درجة الطالب.**
* **ومن ثم قم بكتابة برنامج يقوم بقراءة هذا الملف.**