

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# Object Oriented Programming

البرمجة الشيئية

عدد الساعات: ٢ نظري + ٢ عملي

المتطلبات السابقة: ١١٢ حسب

المادة: م. نجلاء حسن

# Lecture 10

❖ التعامل مع الملفات:

- الملفات التتابعية Sequential Files

- الملفات العشوائية Random Files

❖ الملفات وتدفق البيانات Files and Streams

# الملف والقرص (File and Disk)

- ملف البيانات هو مفهوم منطقي يشير الى تجمع للبيانات، وكل بيان مكون من عدد من البايت، ويتم تخزين مجموعات من البايت على مواقع مختلفة من القرص.
- كل ملف يأخذ اسم. لكي يستطيع البرنامج أن ينشئ أو يستخدم محتويات الملف، يجب أن يحتوى البرنامج على متغيرات تشير الى الملف وترتبط باسم الملف. أي أن اسم الملف هو الرابطة بين البرنامج والقرص.
- توفر لغة ++C أصناف classes مبنية داخل اللغة لتمثل المعلومات المطلوبة عن المجرى الذي سيتم ربطه مع ملف القرص.

# أنواع ملف البيانات

▶ طريقة تخزين واسترجاع البيانات من ملف البيانات تحدد نوع الملف:

## ١- ملف بيانات متتابع (sequential data file): يتم

استرجاع البيانات من الملف بنفس ترتيب تخزينها. (لاستخراج غالبية السجلات يكون مثالي، اما لاستخراج سجل واحد أو عدد قليل من السجلات فان التناول يكون بطيء نسبياً)

## ٢- ملف بيانات عشوائي (random data file): نحدد

موقع كل سجل يجرى تخزينه في الملف بالنسبة لبداية الملف. بناءً على ذلك نستطيع تناول واستخراج السجل المطلوب من الملف مباشرة دون الحاجة لقراءة أي سجلات أخرى. (يوفر تناول سريع)



# استخدام الصنف istream:

- يختص الصنف istream بأداء المهام المتعلقة بادخال البيانات .
- يوضح الجدول التالي أهم دوال الصنف istream واستخدام كلٍ:

الاستخدام	الدالة
استقبال جميع انواع البيانات الاساسية	>>
استقبال حرف من لوحة المفاتيح وتخزينه بالمتغير ch	get(ch);
استقبال مجموعة من الحروف بحد اقصى MAX وتخزينها داخل سلسلة بيانات str	getline(str,MAX)
استقبال عدد من الحروف من أحد الملفات بحد أقصى MAX	read(str,MAX)
تعيين حجم مؤشر الملف من بداية الملف	seekg()
تعيين حجم مؤشر الملف من المكان Pos داخل الملف ويأخذ المعامل الثاني احدى القيم : ios::beg و ios::cur و ios::end	Seekg(pos,seek_dir)

# استخدام الصنف ostream :

- يختص الصنف ostream بأداء مهام اخراج البيانات المختلفة.
- يوضح الجدول التالي أهم دوال الصنف ostream واستخدام كل:

الاستخدام	الدالة
اخراج جميع انواع البيانات الاساسية	<<
اظهار الحرف ch	Put(ch)
اضافة عدد size من الحروف الموجودة بالمصفوفة str الي أحد الملفات .	write(str,size)
تعيين مسافة مؤشر الملف بالبايت من بداية الملف	Seekp(position)
تعيين مسافة مؤشر الملف من المكان position داخل الملف ويأخذ المعامل الثاني احدى القيم ios::beg و ios::cur و ios::end	Seekp(position,seek_dir)

# فتح الملف والتحرك خلال البيانات

▶ الانتقال الى بداية البيانات (بداية الملف):

`ios::beg:`

▶ الموضع الحالي للبيانات:

`ios::cur:`

▶ الانتقال الى نهاية البيانات (نهاية الملف):

`ios::end:`

# دوال أصناف معالجة الملفات

- ▶ فتح ملف بيانات: الدالة `open( )` مع اسم كائن صنف الملف
- ▶ نهاية ملف البيانات: الدالة `eof( )` مع اسم كائن صنف الملف
- ▶ غلق ملف البيانات: الدالة `close( )`

# العمل مع الملفات

- ▶ تحتاج معظم البرامج الى تخزين البيانات داخل ملفات ثم قراءتها مرة أخرى من هذه الملفات.
- ▶ ويحتاج التعامل مع الملفات الى مجموعة أخرى من الأصناف وهي:

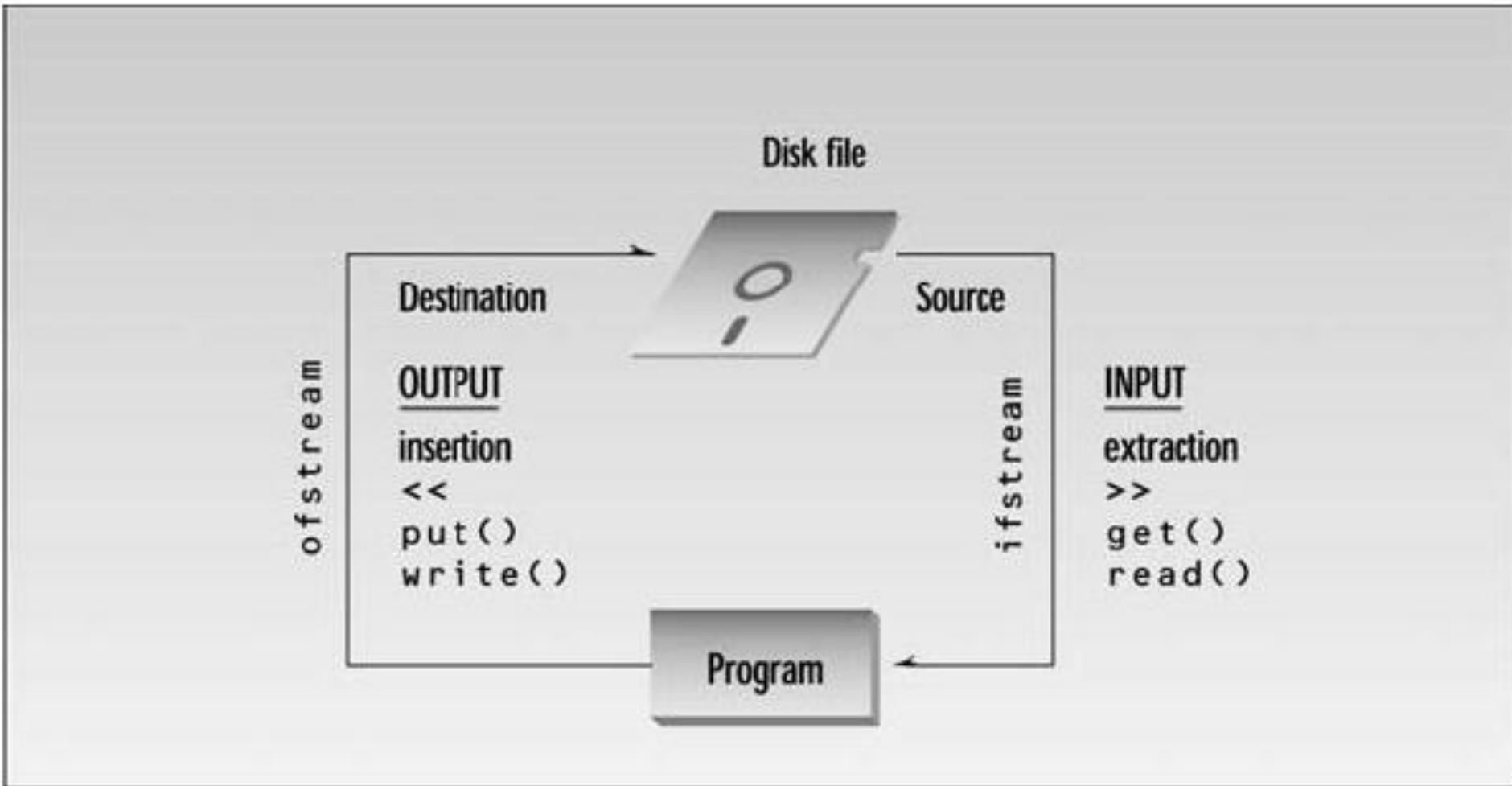
- الصنف `ifstream` تدار بواسطة القراءة من الملف

(و القراءة من لوحة المفاتيح (الادخال) تدار بواسطة `(Cin) (istream object)`.)

- الصنف `ofstream` تدار بواسطة الكتابة في الملف

(و الكتابة تدار بواسطة `(Cout) (ostream object)`.)

- ▶ عند التعامل مع الملفات، يجب تضمين ملفي الترويسة `iostream.h` و `fstream.h` حيث يحتوي الملف `fstream.h` على أصناف الدفع `ifstream` و `ofstream` ، و `fstream` (لإدخال وإخراج بيانات من الملفات



*File input and output.*

# كتابة بيانات الي الملف:

- ▶ عند الكتابة في ملف يجب إنشاء كائن تابع للصف `ofstream`
- الصيغة العامة لاشتقاق كائن من الصف `ofstream`  
(اسم الملف بالقرص) اسم الكائن `ofstream`
- مثال: `ofstream OABC(“\C:test.txt”);`

```
#include<fstream.h>
```

```
void main()
```

```
{
```

```
    char ch='y';
```

```
int j=50;
```

```
ofstream outfile ("\\C:nagla.txt");
```

```
outfile<<j;
```

```
outfile<<" hello world!\n";
```

```
outfile<<" iam nagla\n";
```

```
outfile<<" ";
```

```
outfile.put(ch);
```

```
}
```

صممي برنامج بلغة C++ للكتابة في الملف  
\\C:nagla.txt

## قراءة بيانات الي الملف:

- ▶ يتم فتح الملفات لقراءة بيانات منها بإنشاء كائن يتبع الطبقة ifstream.
- الصيغة العامة لاشتقاق كائن من الصنف ifstream:  
(اسم الملف بالقرص) اسم الكائن ifstream  
مثال: ifstream ABC(“\C:test.txt”);

## مثال:

- ▶ صممي برنامج بلغة ++C للقراءة من الملف \C:nagla.txt

```

#include<fstream.h>
void main()
{
const int max=80;
char ch;
char str[max];
ifstream infile ("\\C:nagla.txt");
//infile>>ch>>str;
//cout<<ch<<str;
infile.getline(str,max);
cout<<str<<endl;
while (!infile.eof())
{
infile.get(ch);
cout<<ch<<endl;
}
/*while(!infile.eof())
{
infile.getline(str,max);
cout<<str<<endl;
} */
infile.get(ch);
cout<<ch<<"\n";
}

```

تكافئ While(infile>>str) { ... }  
while(!infile.eof()) {infile>>str; ....}

# مثال:

```
# include <iostream.h>
# include <fstream.h>
Void main()
{
  ifstream infile("\\D:input.txt");
  ofstream outfile("\\D:output.txt");
  char word[10];
  char c;
  while(infile>>word) // طالما القراءة مستمرة
  {
    outfile<<word;
    infile.get(c) ; // لقراءة الحرف التالي
    outfile.put(c) ;
  }
}
```

صممي برنامج بلغة C++ لقراءة كلمات من  
الملف e:\input.txt ثم كتابة الكلمات في  
الملف e:\output.txt

# تتاول ملف عشوائي

- ▶ 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 نرغب 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.
- ▶ عملية تتاول بيانات ملف عشوائي يلزمها التحكم في المؤشر الذي يشير الى بيانات الملف. عندما نفتح الملف يقف المؤشر عند بداية الملف.

# تتاول ملف عشوائي

▶ نستطيع وضع المؤشر عند الموضع أو البيانات المطلوب قراءتها بتحرك المؤشر أو إزاحته الى الموضع المطلوب باستخدام الدالة المناسبة من الدوال الأعضاء في الأصناف السابقة.

▶ الدوال هي:

- seekg مع صنف القراءة من الملف ifstream

- seekp مع صنف الكتابة (التخزين) ofstream

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

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

٢ / مع وسيطتين الأولى إزاحة من موقع معين في الملف والثانية الموقع الذي تم قياس الإزاحة :  
(النقطة النسبية(الصفرة)،الإزاحة بالبايت)seekg  
(النقطة النسبية(الصفرة)،الإزاحة بالبايت)seekp

▶ فمثلاً العبارة :-

```
seekp( -10,ios:: end);
```

▶ ستضع مؤشر الوضع 10 بايتات قبل نهاية الملف.

# Random Files

Here are some examples:

```
inf.seekg(14, ios::cur);    // move forward 14 bytes
inf.seekg(-18, ios::cur);  // move backwards 18 bytes
inf.seekg(22, ios::beg);   // move to 22nd byte in file
inf.seekg(24);             // move to 24th byte in file
inf.seekg(-28, ios::end);  // move to the 28th byte before end of the file
```

Moving to the beginning or end of the file is easy:

```
inf.seekg(0, ios::beg);    // move to beginning of file
inf.seekg(0, ios::end);   // move to end of file
```