بسم الله الرحمن الرحيم

Object Oriented Programming

البرمجة الشيئية

عدد الساعات: ٢ نظري + ٢ عملي

المتطلبات السابقة: ٢١١حسب

أستاذة/المادة: م. نجلاء حسن

Lecture 3

- دوال البناء والهدم
- دالة بناء ناسخةdefault copy constructor
- تضمين كائنات صنف في صنف آخر
- تمرير الكائنات كمعاملات للدالة
- ارجاع الدوال للكائنات

مقدمة:

برنامج يقوم بتعريف كائنين مع تخصيص قيمة الأول من خلال الدالة set_values () والثاني من خلال لوحة المفاتيح.


```
  #include<iostream.h >
   #include<string.h >
 class student
 {
private:
int no;
char name[50];
public:
void set_values(int a,char *b)
{
no=a; strcpy(name,b);
}
void get_values()
{
cout<<"enter no"<<"\t";   cin>>no;
cout<<"enter name";       cin>>name;
}
void show( ) { cout<<no<<"\t"<<name<<"\n"; }
};
void main(  )
{
student S1,S2;
S1.set_values(110,"mona");
S2.get_values();
cout<<"student 1:";S1.show(  );
cout<<"student 2:";S2.show(  );
}
```

- يحتوي المثال السابق على طريقتين لتخصيص القيم لعناصر بيانات الصنف : تعيينها مباشرة أو من المستخدم.
- لكن من الأفضل احيانا ان يقوم الكائن بتعيين القيم الابتدائية لعناصر بياناته بمجرد انشائه دون الحاجة الى استدعاء اي من الدوال الأعضاء، وذلك من خلال دالة خاصة تسمى دالّة البناء (الانشاء) constructor

دوال البناء constructors

- دالة البناء تحجز مكانا في الذاكرة لكائن يمكن اشتقاقة من صنف (طبقة)معينة، ونعرف دالتين هما:
- **دالة بناء كائن ذو بيانات أولية (default constructor):**

(بدون وسائط)

- يستخدمها البرنامج آلياً لبناء كائن فارغ من أي بيانات (مثل الأمثلة السابقة) أو لبناء كائن وملئه ببيانات أولية default values .في الحالتين يمكن ملأ الكائن ببيانات جديدة خاصة به.وكذلك لانشاء مصفوفة من الكائنات الفارغة من أي بيانات.

- **دالة بناء كائن ذو بيانات محددة (constructor):**

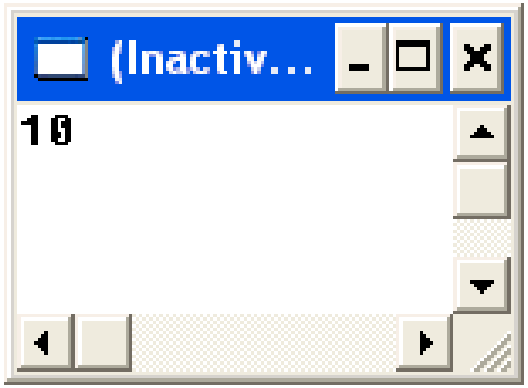
(ذات وسائط)

- يستخدمها البرنامج آلياً لبناء كائن مع ملئه بالبيانات الخاصة به مباشرة (عند اشتقاقه)
- عند اشتقاق كائنات من الصنف يصاحب الدالة اسم الكائن وامامه البيانات المخصصة له.

مثال:دالة بناء بدون وسائط

```
#include<iostream.h >
class newclass
{
  public:
   int x ;
  newclass()
   {
     x=10;
   }
};
void main(  )
{
newclass nc1;
cout<<nc1.x;
}
```

Newclass(){}
Or
newclass(){cout<<“.....”;



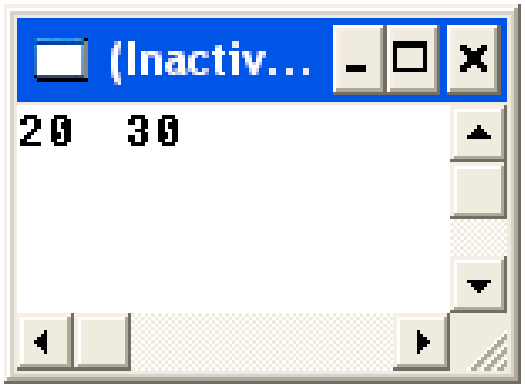
صيغة جديدة لدالة البناء:

Newclass():x(10){}

مثال:دالة بناء ذات وسائط (وسيط واحد)

```
#include<iostream.h >
class newclass
{
  public:
   int x ;
   newclass(int i)
    {
      x=i;
    }
};
void main(  )
{
newclass nc1(20),nc2(30);
cout<<nc1.x<<"  "<<nc2.x;
}
```

يمكن أكثر من وسيط



ما هو الفرق بين دالة البناء والنهج (الدوال)؟؟؟؟؟

دالة الهدم (destructor):

- يستخدمها البرنامج آلياً لحذف أو هدم بيانات كائن من الذاكرة واتاحة الحيز المخصص له بالذاكرة متاحاً لأي برامج أو بيانات أخرى.
- تأخذ نفس اسم الطبقة ويسبقها العلامة ~
- يمكن أن تعرف داخل الطبقةأو خارج الطبقة بوضع اسم الطبقة يليها العامل::

تابع:دالة الهدم destructor

```
class x
{
public:
x( );//     دالة بناء
~x( );//     دالة هدم
};
x::x( )
{
cout<<"constructor is called \n";
}
x::~x( )
{
cout<<"destructor is called \n";
}
```

```
main( )
{
x  x1,x2;
}
```

ناتج التنفيذ....؟؟؟؟

ملاحظات:

1. يتم استدعاء دالة البناء عند اشتقاق الكائن وبصورة تلقائية
2. دالة الهدم يتم استدعاؤها أيضاً بصورة تلقائية ولكن عند نهاية البرنامج
3. كل كائن يستدعي دالة البناء ودالة الهدم

خواص دوال البناء والهدم:

- تحمل نفس اسم الصنف (class)، ولكن دالة الهدم تسبق بعلامة (~)
- يتم تعريفهم في مستوى الحماية Public
- يمكن أن يكون للصنف أكثر من دالة بناء تختلف تبعاً لعدد وسائطها.
- دالة البناء يمكن أن لا تحتوي على وسائط ويمكن أن تحتوي على وسيط واحد أو أكثر.
- يمكن انشاء دالة هدم واحدة فقط.
- ليس لهما أنواع مردود.

استخدام دالة انشاء النسخ (دالة بناء ناسخة)default copy constructor

- لكل صنف أنواع من المشيدات (دوال البناء) على الاقل اثنان:
  default constructor , copy constructor
  X( ); // default
  X(const x&); // copy constructor
- class x s; يستدعى تلقائيا عند تعريف الكائن default constructor
- class x s2(s); يستدعى تلقائيا عند تعريف الكائن copy constructor

هي دالة بناء عادية الا أن المعامل الذي يمرر اليها كائن واحد (الكائن المراد نسخه) و التمرير يكون بالاشارة و من النوع const. x(const x &)

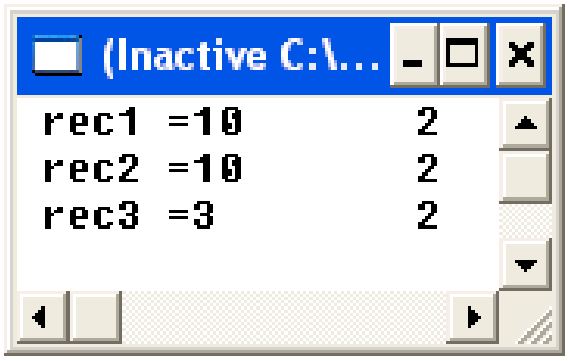
- تستخدم عندما نتعامل مع كائنين (او أكثر) من نفس الصنف.
- اذا تعريف الصنف لا يحتوي على دالة بناء ناسخة ظاهرة، فالمترجم يُنشئها أوتوماتيكيا.

عند استدعاء copy constructor فانه ينسخ الكائن كاملا في الكائن الجديد (من نفس النوع).

تابع: استخدام دالة انشاء النسخ (دالة بناء ناسخة)default copy constructor

```
#include<iostream.h >
class rectangle
{
public:
int x,y;
rectangle(int a,int b)
{x=a;  y=b;}

void show(  ) { cout<<x<<"\t"<<y<<endl; }
};
void main(  )
{
rectangle rec1(10,2);
rectangle rec2(rec1);
rectangle rec3(3,2);
cout<<" rec1 =";     rec1.show( );
cout<<" rec2 =";     rec2.show( );
cout<<" rec3 =";     rec3.show( );
}
```



كائن مشتق من صنف معلن عنه داخل صنف آخر
(تضمين كائنات صنف فى صنف آخر)

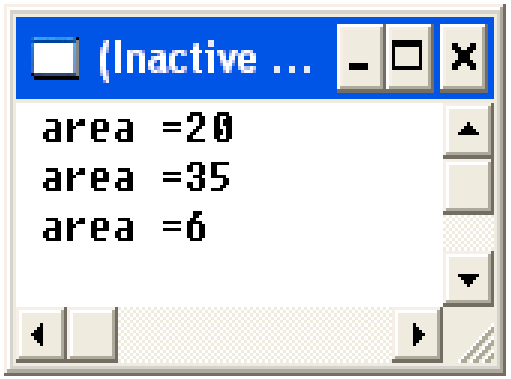
```
#include<iostream.h>
class A{
public:
int a;
A(){}
~A(){}
 };
 class  B
{
 public:
 int b;
 A x;
 B(){};
 ~B(){}
 };
```

```
int main()
{
B  y;
 cin>>y.b;
 cin>>y.x.a;
 cout<<y.x.a;
return 0;
}
```

تمرير الكائنات كمعاملات للدالة:

```
 #include<iostream.h >
 class rectangle
 {
public:
int x,y;
rectangle(int a,int b)
{x=a;  y=b;}

int area(  rectangle ob) { return ob.x*ob.y; }
};
void main(  )
{
rectangle rec1(10,2);
rectangle rec2(7,5);
rectangle rec3(3,2);
cout<<" area ="<<rec1.area( rec1 )<<endl;
cout<<" area ="<<rec2.area( rec2 )<<endl;
cout<<" area ="<<rec3.area( rec3 )<<endl;
}
```



ارجاع الدوال للكائنات:

أكتبي مثال؟؟؟؟؟