

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Object Oriented Programming

البرمجة الشيئية

عدد الساعات: ٢ نظري + ٢ عملي

المتطلبات السابقة: ١١٢ حسب

أ/المادة: م. نجلاء حسن

Lecture 6

-التحميل الزائد للدوال functions overloading

الحاجة الى تحميل الدالات بشكل زائد

▶ اذا افترضنا دالة تحسب مجموع عددين صحيحين:: `int sum(int a,int b);`
العبارة التي تستدعي الدالة تمرر عددين صحيحين كوسائط:: `sum(a,b);`
ماذا يحدث اذا أردنا دالة تحسب مجموع عددين من النوع `float` ؟
`float sum(float a,float b);` وسيكون استدعاء الدالة مشابهاً
وقد تكون دالة تحسب مجموع ثلاث أعداد وهكذا. يجب للمبرمج أن يتذكر كل تلك
الأسماء.

التحميل الزائد للدوال استنساخ الدوال (Function Overloading)

تعريف التحميل الزائد للدوال :

هو إمكانية إنشاء أكثر من دالة تحمل نفس الاسم ولكنها قد تختلف في نوعية المعاملات Parameters Types التي تستخدمها الدالة أو في عدد المعاملات Parameters Numbers ولكل دالة من هذه الدوال الإجراء الخاص بها أي أنه قد تشترك الدوال في اسمها ولكنها قد تختلف في شكلها ومضمونها .

مثال 1:

```
int sum(int a,int b);  
float sum(float a,float b);  
double sum(int a,float b, int c);
```

التعامل مع التحميل الزائد:

لكي يقوم البرنامج باستدعاء دالة sum نقوم بكتابة الاستدعاء بهذا الشكل:
X=sum(y,z);

بناء على نوع بيانات y,z يحدد المترجم أي دالة سوف تستدعي ، كذلك على ترتيب و عدد y,z (ونوع النتيجة التي ترجع بها الدالة).

التحميل الزائد لدوال الطبقة:

اشكاله:

١- التحميل الزائد لدوال

البناء (استنساخ دوال البناء):

ويعني وجود اكثر من دالة بناء بنفس الصنف وتتوقف دالة البناء التي يتم تنفيذها عند انشاء الكائن على عدد ونوع المعاملات المستخدمة في عملية التعريف . كما في المثال.

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

The diagram illustrates function calls from the main function to the constructors of the newclass class. Two arrows labeled 'Call' point from the constructor calls in main to the corresponding constructor definitions in the class. One arrow points from 'newclass nc1;' to 'newclass(){}', and another points from 'newclass nc2(20);' to 'newclass(int i)'. A third arrow points from the 'newclass(int i)' definition to the 'newclass()' definition, indicating that the compiler will use the parameterized constructor for the first call as well.

تابع: التحميل الزائد لدوال الطبقة:

٢- التحميل الزائد لدوال المعالجة.

مثال: ثلاث دوال باسم sum يختلفان
نوع وعدد المعاملات

```
class summation
{
public :
int a, b, c ;
float x, y ;
int sum ( int , int ) ;
float sum ( float , float ) ;
int sum ( int , int, int) ;
};
int summation :: sum ( int a , int b )
{
return a+ b ;
}
float summation :: sum ( float x , float y )
{
return x + y ;
}
int summation :: sum ( int a, int b, int c )
{
return a + b + c ;
}
```

```
main ( )
{
summation ob ;
cout << "Enter a, b, c " ;
cin >> ob .a >> ob.b >> ob.c ;
cout <<"Enter x , y " ;
cin >> ob.x >> ob.y ;
cout << " sum1=" << ob.sum(ob.a, ob.b)
<<"\n" ;
cout << " sum2=" << ob.sum ( ob.x , ob.y ) <<
" \n" ;
cout << " sum3 ="<< ob.sum ( ob.a , ob.b ,
ob.c ) << " \n" ;
}
```

```
(Inactive C:\TCWIN45\Bl...
Enter a, b, c 1 3 5
Enter x , y
2.5 3.5
sum1=4
sum2=6
sum3 =9
```

ملاحظة:

استنساخ الدوال لا يسبب ارباك للمبرمج ، لا يتم اخذ نوع البيان العائد من الدالة في الاعتبار في الدوال المستنسخة ، اذا توافق نوع المعاملات في كل من الدالتين الا ان هنالك اختلاف في نوع القيمة الراجعة فان المترجم يعتبر الدالة الثانية اعادة تعريف خاطئ للدالة الاولى.

```
Unsigned int max(int *,int s);  
Extern int max(int *,int s); // خطأ اعادة تعريف
```