جامعة المنارة

كلية الهندسة

قسم الهندسة المعلوماتية

مقرر برمجة 1

جلسة العملي الرابعة

الفصل الثاني 2022-2023

الجلسة الرابعة

التوابع Functions(الجزء الأول)

الغاية من الجلسة

- ✓ قدرة الطالب على تعريف التوابع و استدعاء التوابع ضمن البرنامج الرئيسي .
- ✓ فهم الفرق بين التابع الذي يعيد قيمة و التابع الذي لا يعيد قيمة و طريقة استدعاء كل منهما في البرنامج الرئيسي.
- ✓ تمرير البارامترات ضمن التابع بالقيمة و كذلك تمريرها بالمرجع أيضاً.

مقدمة عن التوابع

التابع هو برنامج جزئي يقوم بتنفيذ مهمة محددة ، و يمكن استدعاؤه في أي مكان في البرنامج الرئيسي .

إن استخدام التوابع في كتابة البرامج يحقق عدة ميزات :

- ✓ تجزئة الكود : يفيد تعريف التوابع في تجزئة الكود إلى وحدات صغيرة تسهل إدارتها و فهمها و التعامل معها .
- ✓ التقليل من تكرار الكود : يسهم استخدام التوابع في التقليل من تكرار الكود في أماكن مختلفة في البرنامج.
- ✓ إعادة الاستخدام : تستخدم التوابع لتحقيق إعادة الاستخدام لنفس الكود في أماكن مختلفة في البرنامج من خلال استدعاء التابع عدة مرات في البرنامج عند الحاجة إليه .

البرنامج الأول

اكتب برنامج يتم فيه حساب x^y باستخدام المكتبة الرياضية ثم احسب الجذر التربيعي لكل من x ,y .

فكرة الحل :

نحتاج لحساب x^y إلى استخدام تابع القوة pow(x,y) ،وكذلك نحتاج لحساب الجذر التربيعي لـ x إلى استخدام تابع الجذر التربيعي sqrt(x) ، و هي توابع جاهزة للاستخدام معرفة ضمن لغة c++ ، و توجد في الملف الرأسي math.h .

إن مكتبة math تحوي مجموعة من التوابع الرياضية الجاهزة التي يحتاجها المبرمج ، و عندما يراد التعامل مع هذه التوابع يجب تضمين الملف الرأسي math.h باستخدام عبارة #include .

```
#include <iostream>
#include <math >
using namespace std;
 int main()
{
    int x,y;
    cin>>x;
    cin>>y;
```

```
    int power=pow(x,y);
    cout<<"power"<<power<<endl;
cout<<"sqrt(x)"<<sqrt(x)<<endl;
cout<<"sqrt(y)"<<sqrt(y)<<endl;
 return 0;}
```

البرنامج الثاني

اكتب برنامج يتم فيه التصريح عن تابع يعيد مجموع ثلاثة اعداد من النمط float واستخدمه في برنامج رئيسي لطباعة المجموع.

فكرة الحل :

يتم تعريف تابع sum يحوي ثلاثة وسطاء من النمط float تمثل الأعداد الثلاثة المراد جمعها ، و هذا التابع يعيد قيمة حقيقية عبارة عن مجموع هذه الأعداد .

في البرنامج الرئيسي يتم إدخال ثلاثة أعداد صحيحة من قبل المستخدم و من ثم استدعاء التابع sum و تمرير هذه القيم إلى التابع .

يتم استدعاء التابع ضمن عملية إسناد ، بحيث يتم تخزين القيمة التي يعيدها التابع في متحول s من النمط float لتتم طباعتها لاحقاً .

في هذا البرنامج تم كتابة التعريف الكامل للتابع (رأس التابع و جسمه) أعلى البرنامج الرئيسي .

ملاحظة1: التابع الذي لايعيد قيمة يتم استدعاؤه بالاسم فقط مع تمرير القيم المناسبة له كبارامترات .

أما التابع الذي يعيد قيمة من خلال تعليمة return فيمكن استدعاؤه ضمن عملية إسناد ، أو ضمن تعليمة الطباعة ، أو بأية طريقة تتيح الاستفادة من القيمة التي يعيدها التابع.

ملاحظة2: توجد طريقتين لتعريف التابع :

- الطريقة الأولى : يتم كتابة التعريف الكامل للتابع (رأس التابع و جسمه) أعلى البرنامج الرئيسي .
- الطريقة الثانية : نكتب رأس التابع أعلى البرنامج الرئيسي ، و بعد ذلك نكتب التعريف الكامل للتابع(رأس التابع و جسمه) أسفل البرنامج الرئيسي .

```
# include <iostream >
using namespace std ;
// التصريح عن تابع جمع الاعداد ويعيد قيمة المجموع
float sum ( float a, float b , float c )
{
float s ;
s =a+b+c;
return (s); // Returns the result
```

```
}
int main ()
{
float  x,y,z,s;
cin>> x>>y>>z;
s=sum ( x,y,z); // Call t he function
cout<<"sum="<<s;
return 0;
}
```

البرنامج الثالث

اكتب برنامج للتصريح عن تابع يعيد القيمة العظمى بين ثلاث قيم صحيحة مدخلة من قبل المستخدم واستخدمه في برنامج رئيسي.

فكرة الحل :

يتم تعريف تابع maximum يحوي ثلاثة وسطاء تمثل الأعداد الثلاثة المراد إيجاد الأكبر بينها ، و هذا التابع يعيد قيمة عبارة عن القيمة العظمى بين الأعداد الثلاثة .

في البرنامج الرئيسي يتم تعريف ثلاثة متحولات صحيحة و إدخال قيمها من قبل المستخدم و من ثم استدعاء التابع maximum و تمرير قيم المتحولات المدخلة إلى التابع، و هنا تم استدعاء التابع ضمن عملية إسناد لتخزين القيمة التي يعيدها في متحول صحيح .

في هذا البرنامج تم تعريف التابع من خلال كتابة رأس التابع أعلى البرنامج الرئيسي ، و بعد ذلك كتابة التعريف الكامل للتابع(رأس التابع و جسمه) أسفل البرنامج الرئيسي.

```
#include <iostream>
using namespace std;
int maximum (int , int , int );
int main()
{   int x , y ,z ,MAX;
   cin>>x>>y>>z;
   MAX = maximum(x,y,z);
   cout << MAX;
    return 0;}
```

```
int maximum (int a,int b,int c)
{      if ((a>b) &&(a>c)){return a;}
 else if ((b>a)&&(b>c)){return b;}
 else return c ;}
```

البرنامج الرابع

عدل كتابة البرنامج السابق بحيث يتم الطباعة داخل التابع دون أن يعيد التابع قيمة الأكبر بين ثلاثة أعداد بالشكل التالي:

فكرة الحل :

يتم تعريف نفس التابع السابق بحيث تتم طباعة العدد الأكبر داخل التابع ،بالتالي سيكون الاختلاف في أن التابع لايعيد قيمة، لذلك نكتب عبارة void أمام اسم التابع و نلغي عبارة returnداخل التابع.

استدعاء التابع في البرنامج الرئيسي تكون بذكر الاسم فقط مع تمرير القيم المناسبة ، و لايجوز استدعاء التابع الذي لا يعيد قيمة ضمن عملية إسناد أو عملية الطباعة .

ملاحظة : عندما يتم استدعاء تابع لا يعيد قيمة يجب الانتباه إلى عدم إسناده إلى متحول أو عدم استخدامه ضمن تعليمة الطباعة .

```
#include <iostream>
using namespace std;
void maximum (int , int , int );
int main()
{int x , y ,z;
   cin>>x>>y>> z ;
   maximum (x,y,z);
    return 0;}
void maximum (int a,int b,int c)
{      if ((a>b) &&(a>c)){cout <<a;}
 else if ((b>a)&&(b>c)){cout <<b;}
 else cout<< c ;}
```

انتبه لاختلاف طريقة
الاستدعاء عند استخدام void

البرنامج الخامس

نفذ البرنامجين التاليين حيث تم تعريف التابع swap بطريقتين و لاحظ الفرق بينهما .

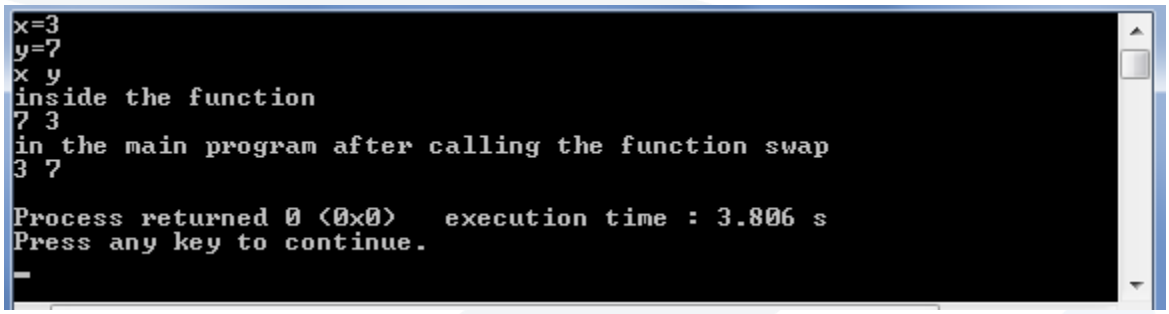
ملاحظة: يمكن تمرير البارامترات ضمن التابع إما بالقيمة أو بالمرجع .

1- التمرير بالقيمة ضمن التابع :

إن تمرير البارامترات بالقيمة ضمن التابع swapping يعني أنه عند استدعاء التابع في البرنامج الرئيسي من أجل قيم x,y يتم إنشاء نسخة عن قيم المتغيرات الفعلية x,y ووضعها في المتغيرات الشكلية a,b الخاصة بالتابع و يحصل العمل على المتحولين a,b ضمن التابع ، بالتالي التغيير ضمن التابع لايؤثر على المتغيرات الفعلية x,y و لايتم تبديل قيم x,y .

```
#include <iostream>
using namespace std;
void swapping (int , int );
int main()
{  int x , y ;
   cout<<"x=";
   cin>>x ;
   cout<<"y=";
   cin>>y;
   cout<<"x y"<<endl;
   swapping  (x,y);
   cout<<"in the main program after calling the function swap\n";
   cout<< x<<" "<<y<<endl ;
    return 0;}
void swapping (int a, int b){
int temp ;
temp = a ;
a=b ;
b = temp ;
cout<<"inside the function\n";
cout<< a<< " " << b <<endl;}
```

الخرج


x=3
y=7
x y
inside the function
7 3
in the main program after calling the function swap
3 7
Process returned 0 (0x0)   execution time : 3.806 s
Press any key to continue.