كلية العلوم

القسم : **الرياضيات**

السنة : **الثانية**

المادة : لغات البرمجة٢

المحاضرة : الثالثة/عملي/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر(What's app-Telegram)على الرقم 0931497960

الجمهورية العربية السورية

جامعة طرطوس

كلية العلوم قسم الرياضيات

السنة الثانية

المادة: لغات برمجة2 _ عملي

المحاضرة الثالثة

Function

التوابع

التوابع(function): هو عبارة عن وحدة مستقلة عن كود البرنامج مصممة لإنجاز مهمة محددة.

يوجد نوعين من التوابع بلغة c++:

1. التوابع الجاهزة المتوفرة في المكتبة المعيارية للغة C++ حيث يجب أن نستدعي Math.h لإنجاز الحسابات الرياضية العادية.
2. التوابع المكتوبة من قبل المبرمج

طريقة كتابة التابع:

1. يتم التصريح عنه أولاً في حال أردنا كتابته تعريفه بعد Main.
2. يتم كتابة تعريف التابع قبل main ويمكن في هذه الحالة الاكتفاء بالتعريف دون الحاجة للتصريح.

تعريف التابع:

```
Return_type    function_name (parameter)
{
 statement;
}
```

حيث :

Return_type: هي نوع القيمة المعادة

- في حال كان التابع يعيد قيمة يكون النوع هو نوع هذه القيمة (float, int, double char,) وتعاد هذه القيمة بواسطة return.
- في حال كان التابع ينجز عمل دون إرجاع قيمة يكون نمط التابع هو void ولا نحتاج return.
- في حال لم يذكر نوع التابع فهو افتراضياً int.

Function_name: اسم التابع وهو اسم اختياري من قبل المبرمج ولكن يجب ألا يكون اسم محجوز.

Parameter: المتغيرات الخاصة بالتابع (الوسطاء) يتم تعريفهن.

Statement: تنفيذ التابع.

نوعه

int

لأنه يعيد قيمة

مثال:

اكتب تابع لحساب جمع عددين صحيحين:

```
Int sum1 ( int x , int y)
{int z;
Z= x+y;
Return z;
}
```

متغيرات التابع العددين الصحيحين

المتغير الذي سوف نحفظ فيه الخرج

نعيد الخرج

مثال2:

تابع طباعة عبارة n hello مرة:

التابع هنا ينجز عمل ولا يعيد قيمة

لذلك النمط void ولا نحتاج return

```
Void print ( int n)
{ for ( int i=0 ; i< n ;i++)
{cout <<"hello";}
}
```

استدعاء التابع:

يتم استدعاء التابع بالشكل: function_name(parameter name);

أي يتم استدعاء التابع عن طريق اسمه واسم المتغيرات.

مثال: اكتب برنامج يقوم باستدعاء تابع طبعة عبارة n hello مرة؟

```
#include<iostream.h>
Void print (int n)
{ for ( int i=0 ; i< n ;i++)
{cout <<"hello";}
}
Main()
{ int a;
Cin>>a;
Print(a);
}
```

سطر استدعاء التابع

أنواع إمرار الوسطاء إلى التابع:

يوجد نوعين لإمرار الوسطاء للتابع هما (إمرار بالقيمة – إمرار بالمرجع(العنوان)).

1.إمرار بالقيمة:

المتغير الدائم (ضمن main) يقوم بإمرار قيمته فقط إلى المتغير المؤقت (ضمن التابع)

لذلك أي تغيير على المتغير المؤقت لا تؤثر على المتغير الدائم.

مثال:

```
Void swap (int x, int y)
{
int z;
Z=x;
X=y;
Y=z;
}
Int main()
{ int a=2, b=5;
Swap (a,b);
Cout<<a<<"\n";
Cout<<b<<"\n";
}
```

الخرج سوف يكون
2
5

المتغير a يمنح قيمته للمتغير x والمتغير b

منح قيمته للمتغير y

في هذا السطر تم استدعاء تابع التبديل

وتم التبديل بين قيمة y و x

أي أصبحت قيمة x=5 وy=2

ولكن هذا التغيير لا يؤثر على a وb

2.إمرار بالعنوان (المرجع):

المتغير الدائم (ضمن main) يقوم بإمرار قيمته وعنوانه إلى المتغير المؤقت (ضمن التابع)

لذلك أي تغيير على المتغير المؤقت تؤثر على المتغير الدائم.

يكون الأمرار بالمرجع بالشكل: parameter_type & parameter_name

مثال:

```
Void swap (int & x, int & y)
{
int z;
Z=x;
X=y;
Y=z;
}
Int main()
{ int a=2, b=5;
Swap (a,b);
Cout<<a<<"\n";
Cout<<b<<"\n";
}
```

الخرج سوف يكون
5
2

المتغير a يمنح قيمته وعنوانه في الذاكرة للمتغير x والمتغير b منح قيمته وعنوانه للمتغير y في هذا السطر تم استدعاء تابع التبديل وتم التبديل بين قيمة y و x أي أصبحت قيمة x=5 و y=2 وهذا التغيير يؤثر على a و b فتصبح a=5 و b=2

ملاحظة: عندما يتم ذكر كلمة دالة في السؤال فهذا يعني أن التابع يعيد قيمة وعندما يذكر إجرائية فهذا يعني أن التابع لا يعيد قيمة(void).

- عندما نريد من الأجرائية أن نحصل منها على قيمة نجعل هذه القيمة أحد وسطاء التابع ونوع إمرارها هو إمرار بالمرجع.

مثال:

دالة حساب مجموع عددين

```
Int sum1 ( int x , int y)
{int z;
Z= x+y;
Return z;
}
Main ()
{int a,b;
Cin>>a>>b;
Cout<<sum(a,b);
}
```

إجرائية حساب مجموع عددين

```
void sum1 ( int x , int y, int&z)
{
Z= x+y;
}
Main ()
{int a,b,c;
Cin>>a>>b;
sum(a,b,c);
Cout<<c;
}
```

مثال2:

دالة لإيجاد العدد الأكبر بين ثلاثة أعداد

```
Int max1 (int x, int y, int z)
{int m;
If(x>y&& x>z)
m=x;
else if (y>x && y>z)
m=y;
else
m=z;
return m;}
Main ()
{int a , b , c;
Cin>>a>>b>>c;
Cout<<max1(a , b , c);
}
```

إجرائية لإيجاد العدد الأكبر بين ثلاثة أعداد

```
void max1 (int x, int y, int z, int &m)
{
If(x>y&& x>z)
m=x;
else if (y>x && y>z)
m=y;
else
m=z;}
Main ()
{int a , b , c ,d;
Cin>>a>>b>>c;
max1(a , b , c , d);
cout<<d;
}
```

مكتبة A to Z