# البرمجة و الخوارزميات 2

الدكتور: سمير جعفر

المحاضرة : الخامسة

التاريخ :١/١١/٢٠١٦

إعداد : فهمي القاضي && محمد فليون

Web: www.syriamath.net group: Improve our mathematics

سنتكلم في محاضرتنا عن موضوع مهم وهو البنى (structures).

## البنية

هي أداة تمكننا من تعريف أنواع جديدة مركبة في لغة C++

### ➢كيف يمكننا تعريف بنية ؟

الشكل العام للبنية في C++ هو :

*struct* اسم البنية {

نوع الحقل الاول اسم الحقل الاول ;

.

.

.

نوع الحقل الاخير اسم الحقل الاخير ;

};

**مثال:**

نريد تعريف نوع جديد هو النوع العقدي .

*struct complex* {

*double r*; القسم الحقيقي للعدد العقدي

*double i*; القسم التخيلي للعدد العقدي

};

### ➢كيف نستطيع التصريح عن متحولات من هذا النوع الجديد :

*complex c,w*;

حيث $i,r$ من أجل $c$ تختلف عن $i,r$ من أجل $w$ (يعني مركبات(c)اعداد تختلف عن مركبات (w))

للوصول إلى محتويات المتحول $c$ نستخدم (.) كما يلي:

$$c.r;$$

$$c.i;$$

حيث $c.i$ أو $c.r$ اعداد حقيقية تحقق خواص العدد الحقيقي في C++

**مثال :**

**اكتب برنامج يقوم بإدخال عددين عقديين وطباعة :**

١- مرافق كل منهما .
٢- ناتج جمعهما .
٣- ناتج طرحهما .

```
#include < iostream.h >
struct complex {
double r;
double i;};
int main (){
complex  x,y;
cin >> x.r >> x.i;
if (x.i == 0)cout << x.r << endl;
 else if (x.i > 0)cout << x.r <<' −'<< x.i <<' i' << endl;
else cout << x.r <<' +' << −x.i <<' i' << endl;
```

قمنا بتعريف نوع جديد وهو النوع العقدي

القسم الاول

**شرح القسم الاول :** قمنا بتعريف عددين (x)و(y) من النوع complex (العقدي) ادخلنا مركبات x من قسم حقيقي(x.r) وقسم تخيلي (x.i)وبعدها ناقشنا حالات القسم التخيلي للعدد العقدي إذا كان $(x.i == 0)$ اي ان القسم التخيلي يساوي الصفر عندها يكون المرافق يساوي القسم الحقيقي فلن يطبع إلا القسم الحقيقي أما إذا كان $(x.i > 0)$ عندها يطبع القسم الحقيقي وإشارة السالب (حيث المرافق ينتج بعكس إشارة القسم التخيلي للعدد) ويطبع القسم التخيلي مع $(i)$ حيث ندخل $i$ لتدل على القسم التخيلي أما الحالة الاخيرة لـ else وهي$(x.i < 0)$ نفس الكلام ينطبق على حالة$(x.i > 0)$ لكن غيرنا الإشارة.

إذا دخلنا العددين ٢و٤ يظهر على شاشة التنفيذ

```
#include <iostream>
using namespace std;
struct complex {
double r;
double i;
};
int main (){
complex  x,y;
cin>>x.r>>x.i;
if (x.i==0)cout<<x.r<<endl;
else if (x.i>0)cout <<x.r<<"-"<<x.i<<"i"<<endl;
else cout<<x.r<<'+'<<-x.i<<"i"<<endl;
return 0;
}
```

```
C:\Users\mohmmad\Desktop\n\...
4
-3
4+3i

Process returned 0 (0x0)   execution time
: 4.220 s
```

**ملاحظة** العدد الذي ليس له إشارة هو عدد موجب.

**القسم الثاني**

```
cin >> y.r >> y.i;
if (y.i == 0)cout << y.r << endl;
else if (y.i > 0)cout << y.r << '-' << y.i << ' i' << endl;
else cout << y.r << '+' << -y.i << ' i' << endl
```

نفس الشرح للقسم الاول لكن هنا قمنا بإدخال مركبات العدد y

كما نرى على شاشة التنفيذ.

```
#include <iostream>
using namespace std;
struct complex {
double r;
double i;
};
int main (){
complex  x,y;
cin>>y.r>>y.i;
if (y.i==0)cout<<y.r<<endl;
else if (y.i>0)cout <<y.r<<"-"<<y.i<<"i"<<endl;
else cout<<y.r<<'+'<<-y.i<<"i"<<endl;
return 0;
}
```

```
C:\Users\mohmma...
5
-2
5+2i

Process returned 0 (0x0)   execut
ion time : 5.814 s
Press any key to continue.
```

```
complex sum;
```

**الطلب الثاني:**
**وهو ناتج جمع عددين عقدين**

$sum.r = x.r + y.r;$

$sum.i = x.i + y.i;$

$if\ (sum.i == 0) cout << "sum = " << sum.r << endl;$

$else\ if\ (sum.i > 0) cout \ll \text{sum=} \ll sum.r \ll' +'$

$<< sum.i << 'i' << endl;$

$else\ cout \ll \text{sum=} \ll sum.r \ll sum.i \ll' i' \ll endl;$

**شرح الطلب الثاني :** قمنا بتعريف متحول $sum$ من النوع $complex$

وقمنا بجمع العدد الحقيقي لـx مع العدد الحقيقي لـ y واسندناهم إلى القسم الحقيقي للمتحول $sum$ نفس العملية بالنسبة للقسم التخيلي

إذا كان $(sum.i == 0)$ اي ان مجموع القسم التخيلي يساوي الصفر يطبع لنا فقط القسم الحقيقي للمجموع أما إذا كان $(sum.i > 0)$ فيقوم بطباعة القسم الحقيقي للمجموع وهنا وضعنا $'+'$ لان لغة c++ لا تضع إشارة للعدد السالب .والقسم التخيلي مع $i$ لدلالة على التخيلي.

لنجمع العددين التالين؟ $2 + 6i + (-1) - 4i =$

```
#include <iostream>
using namespace std;
struct complex {
double r;
double i;
};
int main (){
complex x,y;
cin>>x.r>>x.i;
cin>>y.r>>y.i;
complex sum;
sum.r=x.r+y.r;
sum.i=x.i+y.i;
if (sum.i==0)cout<<"sum="<<sum.r<<endl;
else if (sum.i>0)cout<<"sum="<<sum.r<<'+'<<sum.i<<'i'<<endl;
else cout<<"sum="<<sum.r<<sum.i<<'i'<<endl;
return 0;}
```

```
Select C:\Users\mohmm...
2
6
-1
-4
sum=1+2i

Process returned 0 (0x0)   execution
 time : 67.581 s
Press any key to continue.
```

$complex\ mult;$

$mult.r = x.r - y.r;$

**الطلب الثالث:**
**وهو ناتج طرح عددين عقدين**

$mult.i = x.i - y.i;$

$if(mult.i == 0)cout << "mult = " << mult.r << endl;$

$else\ if\ (mult.i > 0)cout << "mult = " << mult.r << ' + ' <$
$< mult.i << 'i' << endl;$

$else\ cout << "mult = " << mult.r << mult.i << 'i' << endl;$

## شرح عن الطلب الثالث

عرفنا متحول $mult$ من النوع $complex$ نفس عملية الجمع لكن هنا طرحنا.

لنطرح العددين التاليان $(8+1i)-(2+2i)$ =?

```
#include <iostream>
using namespace std;
struct complex {
double r;double i;
};
int main (){
complex x,y;
cin>>x.r>>x.i;
cin>>y.r>>y.i;
complex mult;
mult.r=x.r-y.r;
mult.i=x.i-y.i;
if(mult.i==0)cout<<"mult="<<mult.r<<endl;
else if (mult.i>0)cout<<"mult="<<mult.r<<'+'<<mult.i<<'i'<<en
else cout<<"mult="<<mult.r<<mult.i<<'i'<<endl;
return 0;}
```

```
C:\Users\m...
8
1
2
2
mult=6-1i

Process returned 0 (0x0)
execution time : 25.895 s
Press any key to continue.
```

Syria Math

طلب إضافي
وهو ناتج ضرب
عددين عقدين

$complex\ n;$

$n.r = x.r * y.r;$

$n.i = x.i * y.i;$

$if(n.i == 0)cout << "n = " << n.r << endl;$

$else\ if\ (n.i > 0)cout \ll \text{n=} \ll n.r << ' + ' \ll n.i << ' i' << endl;$

$else\ cout << "n = " << n.r << n.i << 'i' << endl;$

$return\ 0;\}$

**و نكمل هكذا**

**تمرين :**

**اكتب دالة لطباعة عدد عقدي .**

$$void\ print\ (complex\ c)\{$$

$$if(c.i == o)cout \ll c.r;$$

$$else\ if\ (c.i > 0)\ cout \ll c.r \ll' +' \ll c.i \ll' i';$$

$$else\ cout \ll c.r \ll c.i \ll' i';$$

$$cout \ll endl;\ \}$$

$complex * p;$ مؤشر يؤشر على متحول من النوع العقدي .

$p = \&c;$

حيث $c$ هو عنوان ذاكرة وفي الواقع هو مؤشر على أول خانة من الذاكرة حجزت لكل من الجزأين الحقيقي والتخيلي .
معامل المحتوى يصلح مع المتحولات البسيطة فإذا اردنا محتوى العدد العقدي نتبع ما يلي :

حيث $c.i = 5; c.r = 7;$

$$cout \ll p \longrightarrow r; \implies 7$$

$$cout \ll p \longrightarrow i; \implies 5$$

حيث كتابة السهم على الحاسب هي اشارة ناقص متبوعة بإشارة اكبر .

**تمرين:** اكتب برنامج يقوم بإدخال سجلات $n$ طالب حيث $n$ عدد طبيعي مدخل ثم يقوم بما يلي:

١. طباعة سجل الطالب الحاصل على اعلى علامة
٢. طباعة المتوسط الحسابي لعلامات الطلاب
٣. طباعة سجلات الطلاب الذي حصلوا على أعلى علامة على الأقل من المتوسط الحسابي
٤. طباعة سجلات الطالبات الإناث

علما ان سجل الطالب يتكون من الحقول التالية

| الرقم الجامعي | الاسم | الجنس | العلامة |
|---|---|---|---|

# "انتهت المحاضرة "

## ((تصحيح تمرين مر في المحاضرة الأولى))

أكتب دالة عودية تقوم بحساب قسمة عددين صحيحين.

```
#include <iostream.h>
int func (int x, int y){
 if( y == 0 )   return 0;
  else if( x < y)  return 0;
  else  return (1 + func(x-y, y));}
int main(){
int x,y;
do {cin>>x>>y;}
 while ( x < 0 || y < 0);
cout<<func (x,y);
return 0;
}
```

نعرف التابع (func) والوسيطين (x) و(y)

نقوم بعملية مقارنة للبسط (x) والمقام (y)

في حال كان المقام يساوي الصفر سيكون الناتج لانهاية لذلك سنجعله يعيد قيمة الصفر او نضع بدل (return 0;) نضع (cout<<"Error")

اما في حالة البسط (x) اصغر من المقام (y)سيكون الناتج اصغر من الواحد أي الناتج قيمة حقيقية والمطلوب قيمة طبيعية

اما في اخر عملية تكون ناتج قسمة البسط على المقام وتكرار العملية

نستخدم ال do لادخال القيم الموجبة للأرقام x , y

اما في حال كان احد الارقام المدخلة اصغر من الصفر فانه سيطلب من المستخدم ادخال الرقمين من جديد