```
double d1 = p1 . distanc (p2.x , p2.y);
double d2 = p2 . distanc (p1.x, p1.y);
      System.out.print(p1.distanc(p2));

double distance(point p){
 return (Math.sqrt(Math.pow(x-p.x,2)+Math.pow(y-p.y,2)));
  }
 }// end of point class
```

أو ؛

انتهت

المحاضرة الرابعة عشر

نبدء التذكير بالمحاضرة السابقة .

```
Point p1 , p2 ;
 p1 = new {point( );
```

للتذكير: دالة ←

البانــي : هو طريقة أو دالة لها اسم الصف نفسه ومهمتها بناء أغراض الصف وليس لها نوع ارجاع ويمكن كتابة أكثر من باني ضمن الصف .

ملاحظة :

عند ما لا نقوم (لا يقوم المبرمج) بكتابة باني داخل الصف تقوم اللغة (لغة Java) بتوليد باني عند عملية الترجمة ، يسمى **الباني الافتراضي** يحمل نفس الصف وليس له وسطاء ويقوم باعطاء القيم الافتراضية للمعطيات الاعضاء .

مثال :

①

باني فاضي

```
point( ) {
  x = 0 ;
  y = 0 ;
 }
```

②

```
point (double a){
  x = a ;
  y = 0 ;
 }
```

للنقط التي $y=0$ والـ $x=a$ ،

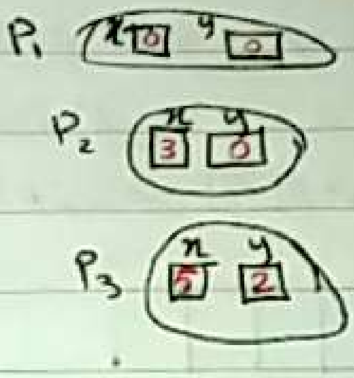
Scanned by CamScanner

/ /

```
(3) point (double a, double b){
        x = a;
        y = b;
        }.
```

مثال:

```
point P1 , P2 , P3 ;
P1 = New point ( ) ;
P2 = New point (3) ;
P3 = New point (5, 2) ;
```



إذا كان مثلاً $P_1.x = 11$; $P_1.y = 21$; فإن هذا الكلام صحيح.

لأنه تم اسناد قيم جديدة

$P_2.x = P_2.x + 2$; | كذلك الكلام صحيح

$P_2.y = P_2.y + 1$; | وهنا تم نقل احداثيات النقطة

ملاحظة: تتعلق بالوسطاء والتعليمة This.

```
point (double x, double y){
    This.x = x ;
    This.y = y ;
    }
```

This كتعريف يمكن أن نقول أنها اسم object للصنف داخل الصنف، أو مؤشر آخر للصنف ضمن الصنف.

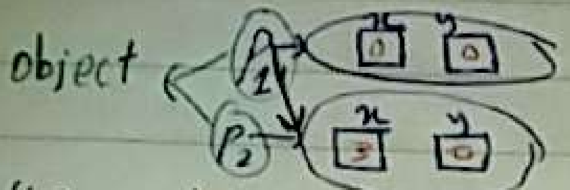
شرح: لو كتبنا:

```
point (double x, double y){
  x = x ;
  y = y ;
  }
```

هنا المترجم لا يستطيع التمييز بين وسيطي الطريقة x,y والمعطيات الأعضاء x,y لذلك استخدمنا التعليمة This سابقاً.

Scanned by CamScanner

// الشرح :

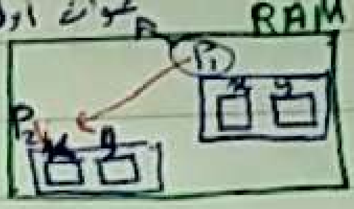


لو قلنا $p = P_1 + P_2$ ليس لها معنى هنا

أما $P_1 = P_2$ هنا عملية إسناد ((يعني مكان يلي اسمو P_2 صار اسمو P_1)).

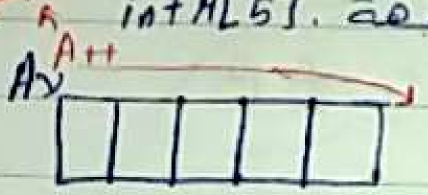
أي :

```
point P1 , P2 ;
P1 = new point (0, 0);
P2 = new point (3, 0);
P1 = P2
```

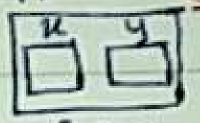


// اسم الـ object هو عبارة عن مؤشر غير واضح بالجافا // تعليق على المؤشرات

int A[5]. بالطريقة (C++)



P++ رح يأشر على أول خانة بعد نهاية الصندوق



وفعلياً P++ لا معنى لها.

مثال 4 :

1) اكتب صفاً يعرف عدد عقدي ويمكن من حساب المرافق وطباعة عدد عقدي بالشكل الديكارتي.

2) اكتب برنامج يقوم بإدخال عدد عقدي وحساب مرافقه وطباعته بالشكل الديكارتي.

الحل : 1) نعمل ملف complex.java

```
class Complex {
  double x ; double y ;
 Complex ( ) { x=0 ; y=0 ; }
 complex (double x, double y) { This.x = x ; This.y = y ; }
   Complex M[illegible] (complex c) {        المرافق :
     Complex r = new Complex ( );
        r.x = c.x;
        r.y = -c.y;
        return r;
        }
```

Void print (complex c){ Void لأن دالة الطباعة لا ترجع شيء

```
if (c.y == 0)   System.out.println (c.x);
else if (c.y >0)   S . O. println (c.x + "+" + c.y + "i");
else   S. O. println (c.x + " " + c.y + "i");
}
```

انتهى الطلب الاول

ملا(2) رح ننشئ ملف اسمو UseComplex.Java للتشغيل كلينا الشغل
ونحط الملف Use complex مع Complex بنفس المجلد منشان الترجمة
أو ممكن أنو نحط كل واحد بمجلد بشرط أنو بس نترجم ملف Complex رح يكون في
بالمجلد يلي هو فيه ملف تاني اسمو Complex.class منسخو للمجلد يلي فيه ملف Use complex

```
class Usecomplex {
public static void main (String[] args) {
    double x; double y;
    x = Stdin.readDouble();
    y =   "     "     " ( );
    complex c = new complex (x, y);
      complex c1 = [redacted] c.M(c);
c.print (c); / أو   c1. print(c1);
    } // end of main
    } // end of Use complex class
```