

題目：

複數運算，利用老師上傳至 My 數位的程式碼修改成完整的複數運算功能。

例如：

ComPx(3,0) => 3

ComPx(0,4) => 4j

ComPx(3,3) => 3+3j

ComPx(3,-4) => 3-4j

.

以此類推

程式碼：

```
def printCompX(c):
    pass

def addCompX(c1,c2):
    return [c1[0]+c2[0],c1[1]+c2[1]]

def subCompX(c1,c2):
    return [c1[0]-c2[0],c1[1]-c2[1]]

def mulCompX(c1,c2):
    return [c1[0]*c2[0]+c1[1]*c2[1]*-1,c1[0]*c2[1]+c1[1]*c2[0]]

def divCompX(c1,c2):
    return float(c1[0]*c2[0]+c1[1]*c2[1])/(c2[0]*c2[0]+c2[1]*c2[1]),float(c1[1]*c2[0]-c1[0]*c2[1])/(c2[0]*c2[0]+c2[1]*c2[1])
```

題目：

將第一題 import 到第二題程式裡，利用物件導向的方式做運算

例如:第一題檔名為 complex.py

```
import complex
```

complex.addcomplex(x, y) → 執行加法運算

complex.subcomplex(x, y) → 執行減法運算

程式碼：

```
def toPyCompX(c):
    return c[0]+c[1]*1j

def testing():
    x=[4,1]
    y=[3,-2]
    px=toPyCompX(x)
    py=toPyCompX(y)
    if toPyCompX(addCompX(x,y))!=px+py:
        print "addCompX(...) failed!"
    else:
        print "Good job! addCompX(...) passed!"
        print toPyCompX(addCompX(x,y))
    if toPyCompX(subCompX(x,y))!=px-py:
```

```
    print "subCompX(...) failed!"
else:
    print "Good job! subCompX(...) passed!"
    print toPyCompX(subCompX(x,y))
if toPyCompX(mulCompX(x,y))!=px*py:
    print "mulCompX(...) failed!"
else:
    print "Good job! mulCompX(...) passed!"
    print toPyCompX(mulCompX(x,y))
if toPyCompX(divCompX(x,y))!=px/py:
    print "divCompX(...) failed!"
else:

    print "Good job! divCompX(...) passed!"
    print toPyCompX(divCompX(x,y))
```

```
if __name__=="__main__":
    testing()
```