

Lecture 02

Lecture contents:

Abstraction data type
Standard Input/Output
Input/Output redirection
Classes and objects
Detailed specs. of ADT

نكل كلام عن ال abstract

الدكتور دخل كتب ال definition ...

ADT : an ADT is a well specified collection of data and a group of operations that can be performed upon the data.

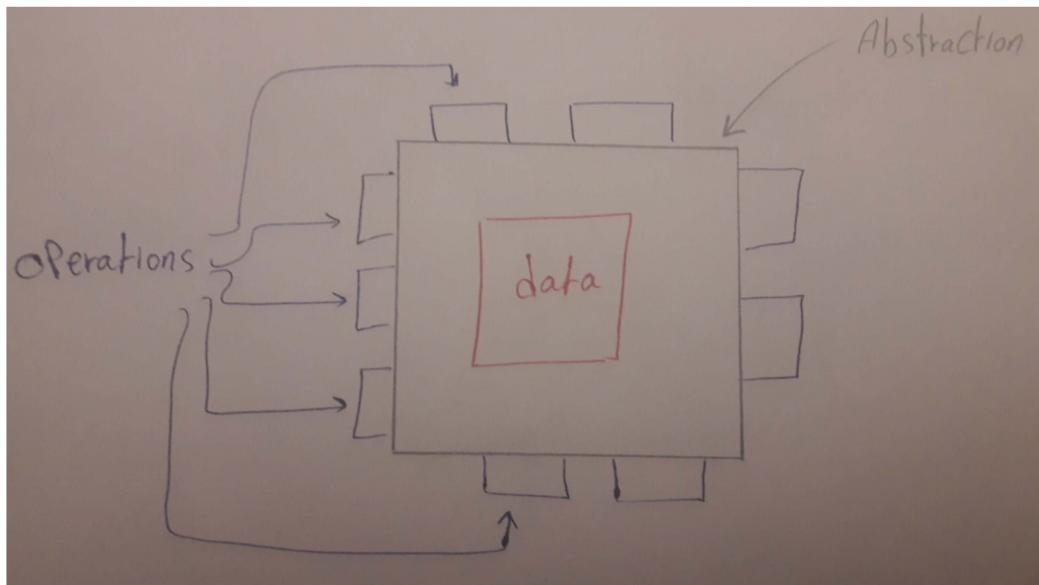
الجملتين الجاين دول نفس المعنى بس بيوضحوا الدنيا اكتر و يفكرونا بالمحاضرة اللي فانت عشان نربط حتى المحاضرتين بعض
Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of value and a set of operations.

The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.

والدكتور كتب

The ADT specifications describes what data can be stored (characteristics) and how it can be accessed **but not how it is implemented or represented in a processor**.

نفس الكلام .. يعني من الآخر عايزين نعمل black box جواه ال data اللي محتاجينها وال operations اللي عايزين نعملها على
الداتا دي .. حاجة زي كدة :



نأخذ مثلاً ... مثلاً الـ

Complex number:

التعريف بباقى الـ "an ADT is a well specified collection of data" ه تكون مواصفاتها ايـه.

Characteristics:

1-Real part.

2-Imaginary part.

: بديهيات D

ما هو مش ده بس المطلوب .. عايزين نقول كمان الأرقام دي هتبقى كام bit عشان بناءً عليه نعرفها في البرنامج .. هي اكيد ممكن تكون سالب وممكن كمان تكون فيها decimal point بباقى يا اما float double

Double  64 bits

Float  32 bits

انت بقى فرر حسب دقة الأرقام اللي انت عايزها و هكذا ... كلام قديم يعني .. المهم كده طبقنا اول جزء في التعريف .. نكمـل باقيـته
كـده ".and a group of operations that can be performed upon the data" اـنا في البرنامج اللي عـايز اكتـبه دـه .. ايـه الـ operations اللي هـعملـها على النوع دـه من الدـاتـا اللي هو الـ complex .. خـلى بالـكـ design

: المهم الـ operations

Operations:

ممكن نقـى عـايزـين نـجمـع رـقمـين complex عـلـى بـعـض بـيـقـى

1-Add.

وهـكـذا بـقـى نـسـأـل نـفـسـنـا مـحـتـاجـين نـعـمـل ايـه في الـ complex numbers دـى ..

2-Subtract.

3-Multiply.

4-Divide.

ويكون كمان لو واحداً بنكتب الـ design هنا نكتب كل واحدة بـتعـمل ايـه ... مثلاً:

5-Get magnitude: calculates the magnitude of the complex number.

6-Get phase

7-Conjugate: calculates the Conjugate of the complex number.

8-Read complex: read real and imaginary from standard input.

دي عـشـان مـثـلاً اـعـمـل cin لو هـسـتـاهـ من الـ user .. بـس ايـه الـ standard input دـى؟؟ .. استـنى شـوـيـة وهـنـعـرـف

9-Set real.

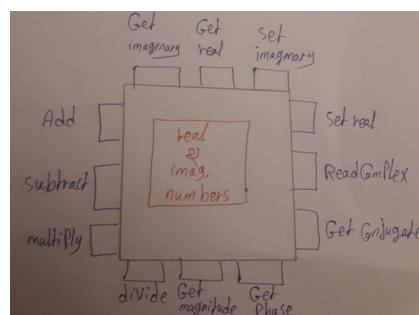
10-Set Imaginary

11-Get real

12-Get imaginary

مـمـكـن في البرـنـامـج بـتـاعـك أـصـلـاً مـتـبـقـاش عـاـيزـ دـه كـلـه وـمـمـكـن تـبـقـى عـاـيزـ حاجـاتـ اـكـتر وـمـمـكـن تـبـقـى عـاـيزـ حاجـاتـ تـانـيـه غـيرـ دـولـ .. فـكـلـ برنـامـج يـخـتـلـف عنـ التـانـيـ.

كـدةـ اـحـنا خـلاـصـ معـاـنـا الـ data اللي هـنـشـتـغلـ عـلـيـهاـ وـمعـاـنـا الـ operations اللي هـنـعـمـلـهاـ عـلـيـهاـ نـقـدـرـ نـمـثـلـهـمـ فيـ الـ black box بتـاعـنا :



تعالى نعرف بقى ايه ال standard input ده؟؟؟ هي حاجة برة موضوع المحاضرة شوية بس هتفيدنا و هتفيدنا في ال assignment كمان.

Standard input and output :

Each running program in windows, linux, android,... operating systems is assigned three default files upon execution.

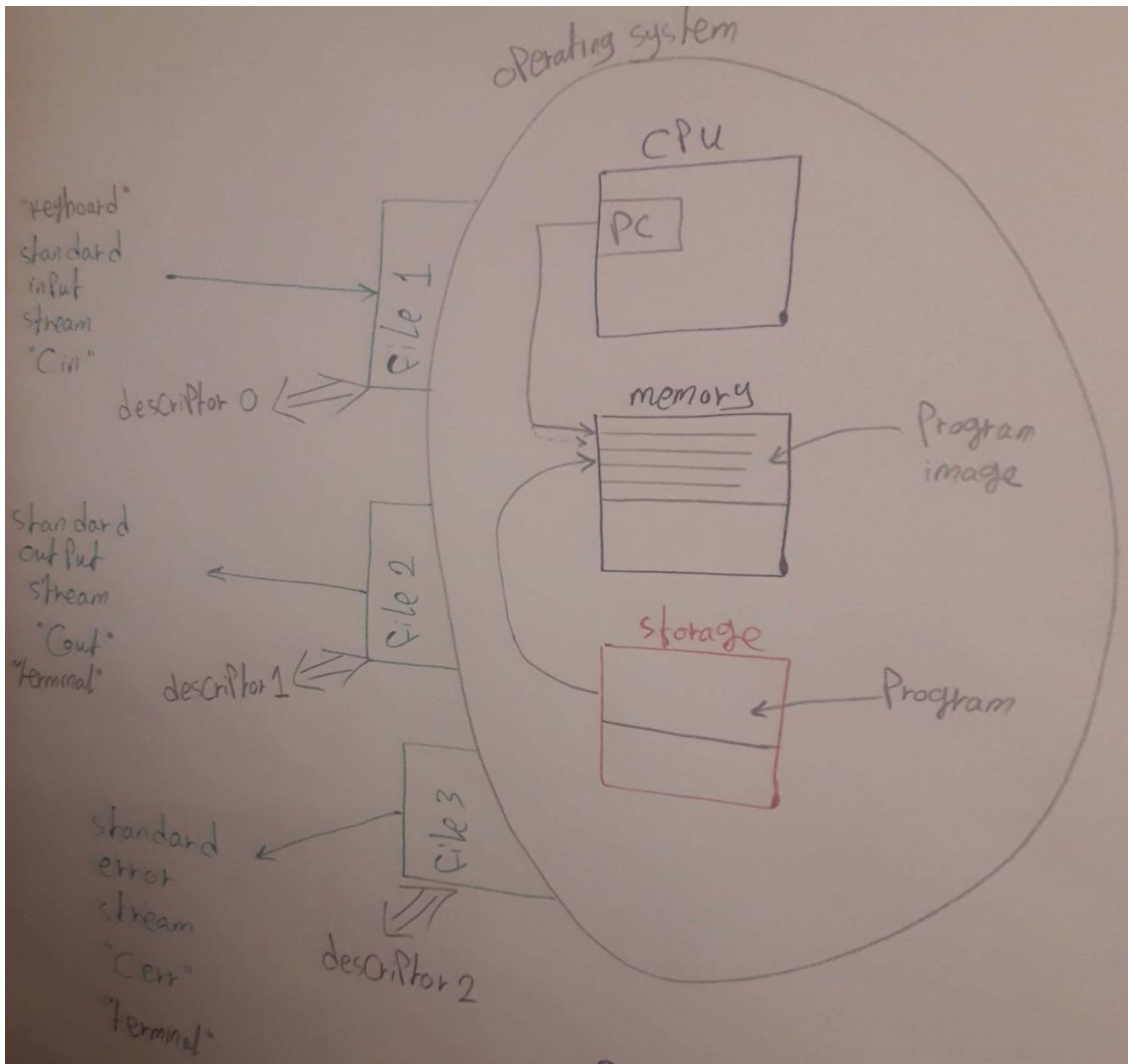
دلوقتى لما هن run ال program وهو موجود في ال storage memory هيبقى فيها ال أصلا فال operating system معين هيـ execute البرنامج ... وهيفتح 3 فايلاـت.

وأحد اسمه **standard input** وواحد اسمه **standard output** يعني مثلاً في البرنامج بتاعك الـ **user** هيكتب الـ **inputs** من الـ **keyboard** وانت هتطلع له الـ **output** على الشاشة قدامه واحتمال تطلع له الـ **errors** ... بقى الـ **file** الـ **standard input** هيأخذ من الـ **keyboard** وانت هتطلع له الـ **file** الـ **standard output** هيطلع على "مش الشاشة بقى" الـ **terminal** ولو هتنظره بقى على الـ **terminal**.

ال descriptor 0 ليه حاجة اسمها standard input وال descriptor 1 ليه standard output وال descriptor 2 ليه standard error ايه ال descriptor ده؟؟

د هر قم بي map من ال files دى للبرنامجه بتاعك.
و ده تعريف لـ descriptor من Wikipedia بيشرح الدنيا اكتر

a file descriptor is an abstract indicator used to access a file or other input/output resource, such as a pipe or network socket.



اعتقد كده أوضح ... التلات فاييلات دول هم ال default في اي برنامج هي
فلو مثلاً كنبت في البرنامج

`Open("mytext","rw")`

هيفتح descriptor جديد غير الثلاثة ال default .
وعلى فكرة من حق اي واحد بيعمل program انه يقفل الثلاثة descriptors دول .. اللي هو مش عايز حاجة من ال user ولا
عايز يوريله حاجة .. واحد مخنوق من نفسه تقريباً: D" لأن الطبيعي انهم بيبقوا موجودين .. بس فيه applications فعلاً مش
محتجة زي مثلاً OS programs .. فال developer بيقفل ال descriptors interaction مع ال user

طيب لو انا مش عايز ال user keyboard يدخلني من ال output او مش عايز اطلعه على ال terminal
يبقى نشوف حاجة اسمها input/output redirection

Input/output redirection:

مثلا

myprog.exe<myinput.txt

كده ال standard input هو ال txt file مش ال keyboard

myprog.exe>out.log

كده ال output يطلع على ال file بدلاً ما يطلع على الشاشة

myprog.exe>&error.log

بردو بطبع ال error على file بدلاً من terminal ... وال & دى بتتحط مع ال redirection بتاع ال error .. وخد بالك بردو لو حاجة . دلوقي السهم بيقى داخل على myprog ولو output السهم بيقى خارج منها.

نتكلم في حاجة تانية بقى في ال programming .. خلي بالك عشان مانتوهش احنا كنا بنتكلم الاول في ال abstraction .. بس خرجنها واتكلمنا عن ال standard input/output وال I/O redirection method دلوقي هنتكلم عن حاجة جديدة في ال programming هي ال class و دي ميزة في C++ مش موجودة في C

Classes and objects:

لو عملنا header file اسمه complex.h .. ال header file دة هنكتبه بعد كدة بس خلينا نعتبر اننا كتبناه وعايزين نكتب ال main code ونحتاج ال #include<complex.h> فنهعمل

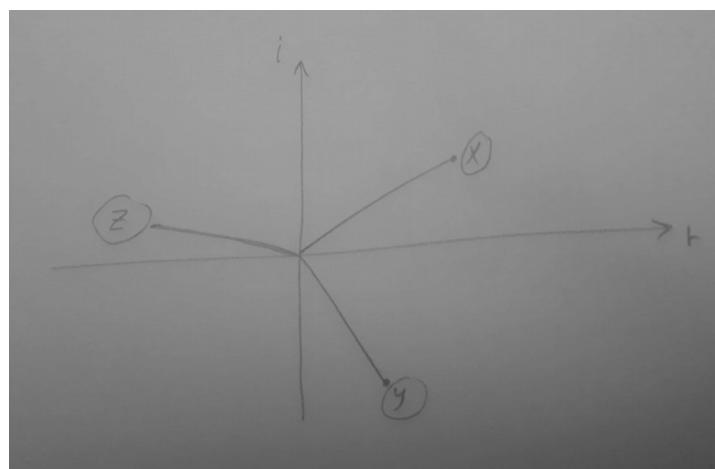
```
#include "complex.h"
```

```
main () {  
    int i,j,k ; //declare 3 variables of type integer (i,j,k)  
    complex x,y,z ; //declare 3 instances (objects) of class (ADT) named "complex"  
}
```

عايزين نركز اكتر في ال class definition header file بتاع ال class اللي اسمه complex x,y,z ... المفروض ان احنا كاتبين في ال header file دة ... دلوقي بقى احنا حجزنا 3 instances (objects) من ال class دة اسمهم x,y,z ... يعني ايه بقى الكلام دة؟؟؟

دلوقي انا انسان .. انا كدة object من ال class الكبير اللي اسمه "بشر" .. بس هل احنا كلنا نفس ال instance او نفس ال object .. لأن انا مفيش حد زبى بالظبط ومفيش حد له نفس بصمتي او بصمة الصوت بتاعتي، ف كل واحد فينا مختلف عن الثاني في صفاتنا بس احنا كلنا فينا صفات مشتركة زي اتنا بنمشي على رجلين مثلاً ولينا نفس الهيئة ... عشان كدة احنا instances مختلفة من class واحد ..

فلو بصينا على x,y,z في الكود هنا ليه ممكن يكونوا مشتركون في حاجات معينة في ال class دة بس هم مختلفين في حاجات تانية عشان كدة كل واحد منهم instance مختلف عن الثاني وله مكان لوحده في الميموري .. عشان الدنيا توضح اكتر تعالى نص على ال representation بتاع x,y,z في ال complex plane وهنلاقي ان كل واحد فيهم مختلف عن الثاني



طيب ايه بقى ميزة ال **instance** عن ال **variable** العادي؟؟
فيه كذا حاجة بس اللي الدكتور اتكلم عنه في المحاضرة هو اننا نقدر نعمل حاجة زي كدة:

`x.ReadComplex();`

`y.Add(x);`

Function call Parameter 'x' which is another instance in the class

ولو فاكر من شوية ان **operations** دي من ال **ReadComplex**, **Add** اللي انكلمنا عنها فوق .. فاحنا نقدر نعمل **operation** او **function** معينة على ال **instance** **i** اللي بتاعنا :

`i.ReadComplex(); XX`

“In general, Operations are mapped mainly as functions (or operators).”

نرجع بقى لـ **abstraction** اللي نسيناه .. دلوقتي عاييزين نعمل **operations** **detailed specification** لـ **operations** بـ **abstraction** حاجة كده تبقى زي **terms & conditions** اللي بنشوفها في فيسيوك او توبيتر بحيث انا لما نقر اها نبقى عارفين كوييس احنا بنتعامل مع ايه وهنكم على مثل ال **complex number** **abstraction** بـ **complex number** اللي بدأناه

Detailed specifications of complex ADT:

ميزة الخطوة دي هنلاقيها بـ **logical errors** فيه **brief description** (او تبقى حاجة زي ضمان) ان الكود مابيقاش فيه **logical errors**

تعالى ناخد مثال زي **function** الجاية دي :

`bool ReadComplex ()`

Return type Function name Function Arguments

عشان اعرف أوصاف ال **function** دي صح بحيث ان اي حد يقدر يقرأ التوصيف دة ويفهم ابعد ال **function** دي فانا في الاول محتاج اكتب **brief description** عن اللي ال **function** دى بـ **Post-conditions** .. بعد كدة هحتاج اكتب 3 حاجات :

- 1-Pre-conditions**
- 2-Post-conditions**
- 3-Return**

1-Pre-conditions

دي باختصار: [الشروط الواجب توافرها عند تنفيذ هذه العملية]

يعني ايه بقى العقد شريعة المتعاقدين والاختصار الحلو اللي فوق دة؟؟ من الآخر كان ال **function** بتقول لل **user** لو اشتغلت بيها صح وحققت شروطي انا هشتغل معاك صح .. بس لو فيه حاجة غلط مستنش مني **output** مظبوط يعني ايه برضه؟؟

دلوقيتي انا معاليا **IC** وفتحت الداتاشيت بتاعها لقيت ال **voltage range** بتاعها من 3.3 ل 5 فولت وجيست انا دخلت 6 volt .. انا مالترمتش بال **pre conditions** اللي الداتاشيت قاللي عليها فماجيش ازعل في الاخر ان ال **IC** احرقت او ماشتغلش صح .. اشطة كدة؟؟

نرجع للديزاین بتاعنا بقى .. ايه الشروط اللي ال **function** محتاجة انها تكون موجودة لما آجي اعمل ال **operation** دي في ال **execution time**؟؟

مبينياً بس عايزين نركز او ي في كلمة **execution time** دي .. يعني مانيفعش اكتب مثلاً اني المفروض اعمل **include** لل **header file** لأن دي حاجة بتتعلّم بدرى واحدنا دلوقي في ال **execution time** يعني خلاص عدinya مرحلة **compilation** بسلام .. فعشان كدة مش هنكتب في ال **pre conditions** اتنا محتاجين ال **#include** وبرضه فيه حاجة مهمة عايزين نأخذ بالنا منها ... بص على الكود دة :

```
main () {
    int i,j,k;
    i = j + k;
    cout << i; }
```

تقدير تعرف قيمة **i** اللي هتطلع على ال **terminal** لأن انا لا عارف قيمة **i** اللي دخلت فيها جوة ال **main** ولا عارف قيم **j** و **k** .. ودة لأن مفيش **initialization** للقيم دي في الأول فاحنا كدة بنعمل **operations** على **garbage** وأكيد دي حاجة ملهاش اي تقسير او **logic**

وبالتالي ك **logic** انا عايز بيقى فيه **condition** ان ال **variables** او ال **instances** اللي هستخدمهم في ال **function** يكونوا **initialized** الاول .. طب يعني بعد كل دة احنا هنكتب ايه في ال **function** بالنسبة لل **pre-conditions** اللي فوق دي؟؟ .. خلينا **skip** الاجابة بتاعت السؤال دة دلوقيتي عشان نشرح النقاطين الباقيين وبعد كدة هنأخذ ال **example** كامل من الأول

2-Post-conditions “The side effects”

هنا بقى هنقول ايه اللي هيحصل في ال **program** بتاعي بعد ما ال **operation** دي تتنفذ .. الموضوع دة هيأثر على مين في باقى ال **program**؟؟

يعني مثلاً تعالى نص على الكود دة :

```
#include "complex.h"
```

```
main () {
int i,j,k ;
complex x,y,z ;
bool condition;
```

```
condition x.ReadComplex (); //apply operation “invoke” named ReadComplex on object ‘x’
}
```

انا كدة عارف ان اخر **statement** دي هتأثر على ال **instance** اللي اسمه **x** ... بس انا بعمل ديزاین حاجة عامة .. مش عارف ال **function** دي هستخدمها **x** ولا **y** ولا حاجة اسمها **bta3** مثلاً .. فعشان اوصفه محتاج اكتب التأثير على الحاجة العامة دي .. اللي هي اسمها **THIS OBJECT** ... طبعاً الكلمة دي غنية عن اي تعريف وصوتها مش هيطلع من وداننا لآخر السنة :

3-Return

ايه ال return بتعاي .. بس احنا عندنا نوعين من ال return بنكتبهم قبل ما نكتب ال int bool ولا دا

ولا ايه :

1-Explicit return:

يترجم حاجة واضحة .. زي int او bool

2-Implicit return:

يترجم حاجة ضمنية .. ممكن الحاجة دي تكون في ال THIS OBJECT او تكون global variable مثلًا

كدة احنا عرفنا ازاي نعمل characterization على الكلمة ونشوف operation لكل examples ازاي ... دلوقتي بقى هنأخذ بيتطبق ازاي .

bool ReadComplex()

Read real and imaginary parts from standard input stream (input file)

Pre-conditions:

-Standard input must be open & has characters representing a proper real and imaginary parts.

عشان ال program يعرف يقرأ
الداتا

يعني لو الفايل فيه حاجة زي كدة :
13.5wu32 + j0.5mk03
دة مش valid complex number .. فكدة انا
ماحققتش ال pre-condition برغم ان ال
مفتوح standard input

Valid complex number is as
following
 $j12.521 + 15.543$

وبرضه فيه precision معين مانقدرش نطلع
عن ال range

Post-Conditions:

-Real and imaginary are stored in THIS OBJECT memory

طب اقدر احط ال touch بتعاي واقول ان فيه condition اسمه close standard output تاني لأن بيقى انا كدة
بوظت الدنيا لأن انا مش من حقي افقل ال standard input على باقي الكود وابهدل الدنيا ال operations اللي جاية بعد كدة ..
فماينفعش احط post-conditions تبهدل الدنيا وانا مش دارسها ودارس تأثيرها كوييس الاول

Return:

True if read succeeds, false if otherwise.

ويبقى انا كدة عملت ال ReadComplex .. لو عرفت أ return value فعلا ال true .. لو حصل مشكلة وال
cerr ماتعملش بقى ال return ب false وساعتها ممكن نعرف ان حصل logical error عن طريق ال read

مثال تاني:

void Add(complex c)

Adds 'c' to THIS OBJECT and puts the result in THIS OBJECT

Pre-conditions:

Both c & THIS OBJECT have been properly initialized ...

"؟؟ garbage

Post-Conditions:

THIS OBJECT is updated with the results of the additions.

Return:

المفروض انها void يعني مش بترجع حاجة .. بس هي في الحقيقة بترجع implicitly لأنها بتغير قيمة ال
Implicitly returns value of addition in THIS OBJECT.

مثال كمان ...

double GetMagnitude ()

Returns the magnitude of THIS complex OBJECT

Pre-Conditions:

THIS OBJECT is properly initialized

Post-Conditions:

None

عادي ممكن يكون فيه simple functions
مش محتاجة post-conditions
و ساعات كمان مش pre-conditions
بتحتاج

Return:

Explicit return the magnitude of complex number represented by THIS OBJECT