

Lecture 03

Lecture contents:

- Setter & getter operations
- Implementation of complex ADT using C++ classes
- Information hiding
- Client program
- Build process
- Discussion

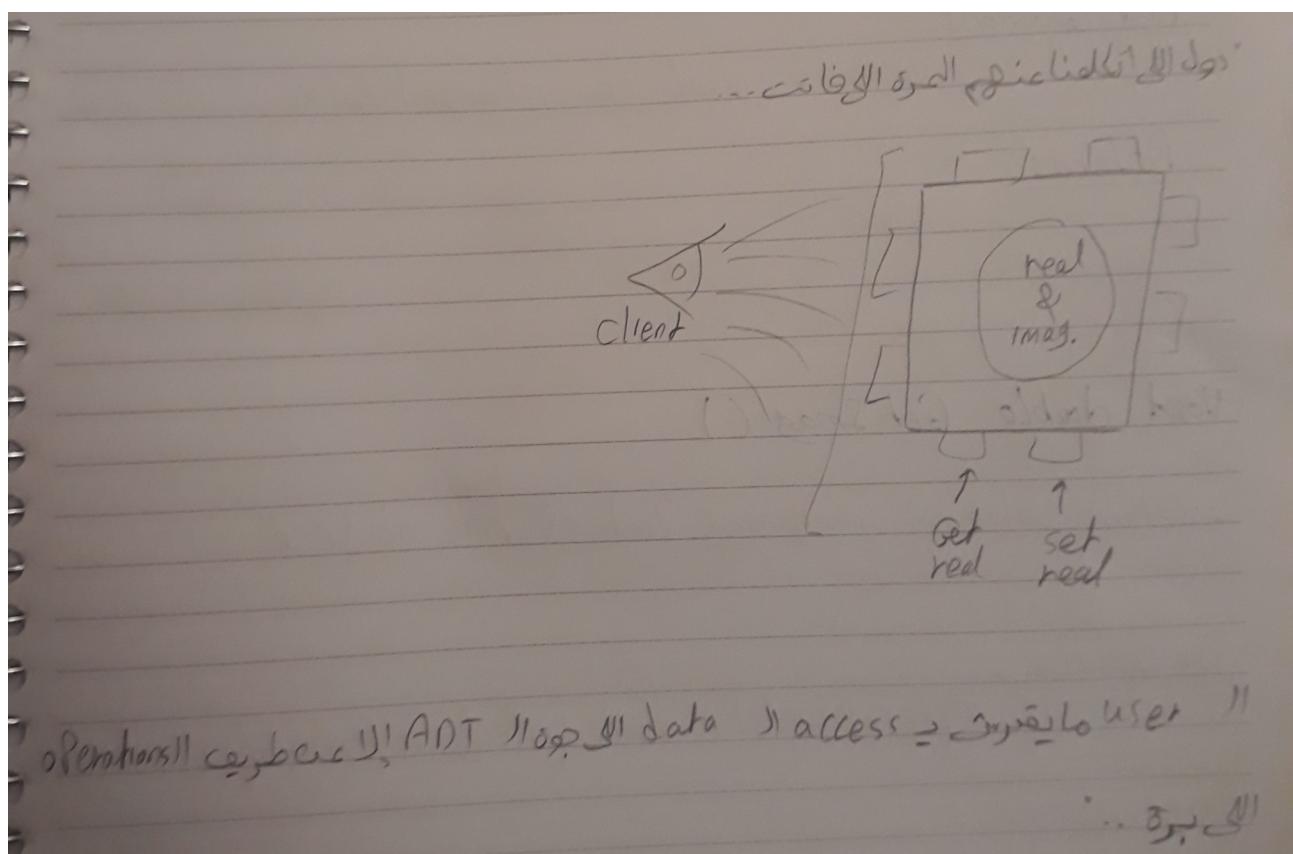
هكمل على المحاضرة اللي فاتت على مثال ال complex number
اول حاجة الدكتور أنكلم عن حاجة اسمها

Setter & getter operations:

عشان ال user يقدر يقدر ال data function فاحنا هنعمله get وبها يقدر يقدر set وهذا ال .. اي بقى ال set وال get .. دول ؟؟

Set operations: allow setting a value of data of an ADT.

Get operations: allow retrieval of values of data of an ADT.



عشان كده عملنا لل user ال get و ال set functions
واحنا عملنا المرة اللي فاتت design لل functions .. تعالوا بقى نعمل ال GetReal()

double GetReal()

Retrieve value of real part of THIS OBJECT

Pre-conditions: Complex object must be properly initialized

هو انا ليه مقولتش هنا ان ال i/p standard لازم يبقى مفتوح ؟؟ عشان هنا انا معايا ال complex number خلاص .. اللي وانا بجيبيه قلت فايل ال i/p يبقى مفتوح لكن هنا انا مش هاخد من ال standard i/p حاجة منا خلاص خدت ال complex number منه وعديت الحنة دى.

Post-Conditions: None.

عشان مش هتأثر في حاجة . ازاي وانا هغير في ال variable اللي هحط فيه ال real ده ؟؟ اديك قلت هتحط في variable يعني ده ال return بناتع ال function

من الآخر .. في جزء ال post-conditions ده انا بكتب الحاجات اللي هتغير غير ال return اللي انا قايل عنه في .double function declaration

Return: value of real part.

Void SetReal(double x)

set value of real part of x.

Pre-conditions: none

لان انا مش بـ read فمش تحتاج ال input stream ولا عايز يبقى في valid complex number قبله.. عشان انا متأكد انها ه تكون موجودة ولو محصلش كان هيحصل compilation error

Post-Conditions: the value of the real part is changed to x.

Return: nothing.

وهذا عشان دى void function فكتبنا الحاجة اللي هتأثر فيها في ال post-conditions مش ال return .
نيجي بقى للكود ... أخيرا D :

Implementation of complex ADT using C++ classes:

دى عباره عن حاجتين

1.class definition.

<ClassName>.h

وده اللي بنعمله الأول

2.class implementation

<ClassName>.cpp

دول فايلين مختلفين عن بعض .. ممكن الفايلين دول يبقوا في فايل واحد بس ده مش recommended

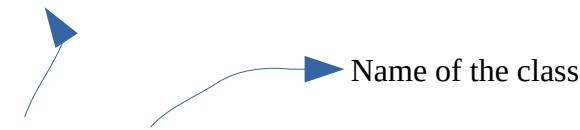
نبدأ بأول فايل

Complex.h

هنا هنعرف ال class complex اللي هو وجواه هنقسم ل حاجتين public و private والمفروض في حاجة كمان اسمها protected بس احنا مش هناخدها

ال public هحط فيها من الآخر ال interface .. الحاجات اللي ال user مسموحله يستخدمها وال private الدكتور قال نحط فيها ال variables اللي جوة ال ADT ومتش عايز ال user يشوفها وقال ممنوع في الكورس بتاعه نحط ال public في ال variables عشان هينقص عليها .

Reserved word in C++
to define a class



```
class Complex{
public:
    double GetReal();
    double GetImag();
    void SetReal(double x);
    void SetImag(double x);
    bool ReadComplex();
    void Add(Complex c);
    void Multiply(Complex c);
    double GetMagnitude();
```

/* هكذا بقى .. كل الـ functions اللي الـ user هيحتاجها او يعني الـ interface اللي في الـ blackbox اللي اكلمنا عنه المحاضرة اللي فاتت.

كل اللي فوق دول اسمهم member functions ولو في حاجة منهم انا بس اللي هستخدمها ممكن احطها في الـ private مثلًا لو عايزيين نعمل GetMagnitude اللي فوق دي .. المفروض اننا بنجيب الـ real² + Imaginary² ونجيلهم function square root كمان اسمها SquareRoot مثلًا .. بس الـ user مش يحتاج يعرفها ولا تحتاج يعرف عنها اي حاجة لأن هو مش هيستخدمها، دة هيستخدم GetMagnitude و GetMagnitude هي اللي هستخدم SquareRoot .. فاحنا نقدر نحط الـ member function اللي اسمه SquareRoot في الـ private عادي */

// نشوف الـ private بقى ..

```
private:
double real;
double imag;
```

/* دول اسمهم member variables ونقول تاني ممنوع منعا باتا نكتب الـ public في الـ variables عشان مدحش يقدر يشوف الـ data دي او يعدل عليها غير عن طريق الـ getter والـ setter مش رخامة .. بس عشان كمان انا اقدر اعمل check وشوف الـ user مدخل data منطقية وللا لا ولو لا يبقى انا اقدر امنعه من كده فالهدف من كل ده هو الـ /* . encapsulation */;

وقفنا قوس الـ class كده احنا عملنا الـ class definition قبل ما نعمل تاني حاجة اللي هي الـ class implementation عايزيين نعرف حاجة اسمها

Information hiding:

the principle that a client program can use an ADT with no knowledge of its implementation is called information hiding.

-A program language that supports information hiding is said to support encapsulation.

يعنى الموضوع يبقى زي الكبسولة ... تخى التفاصيل اللي يبقى واضح كيفية الاستخدام زي ما بنعمل مع الكبسولة .. مش بنقدر نعرف مكوناتها لا بنعرف ناخدها كل دا ليه مثلاً بس.

- احد الـ features القوية في الـ OOP انه بيـ support الـ encapsulation من ضمنه حوار الـ public و الـ private اللي بنقول عليه .. ولسه عنده كلام في اخر المحاضرة يوضح الدنيا اكتر

طيب قبل بردو ما نبدأ في file الـ cpp عارف ايه الفرق بين الـ class و الـ structure ؟؟ في المثال اللي فوق ده انت ممكن تشيل كلمة class و تكتب مكانها structure وهنبقى صح عادي ... او مال ايه الفرق؟

هو فرق صغير اوی .. او استخدمت ال class من غير ما تكتب public و private يبقى كل حاجة private وده طبعاً مينفعش عشان كده محدش هيعرف يستعمله (كأنني عملت IC من غير ال pins بتاعتها) .. دى في حالة انك موضحتش في الكود private وال public ولو استخدمت structure من غير ما توضح وتكتب private و public ... هيبقى كل حاجة public

فزي ما تقول ال default بتاع ال class ببقي private و public ببقي structure ال default

نبدأ بقى نكتب ال class implementation

Complex.cpp

اول حاجة ن include ال file.h

#include "complex.h"

وه كمان ال math.h عشان هحتاج منها حاجات زى ال sqrt قدام زى ما هنشوف

#include<math.h>

: Complex.h كبتاه في ال member function لكل implementation

ولا بنكتب ال declaration اللي كنا كاتبته في ال Complex.h بس قبل اسم ال class هنكتب اسم ال function وال return type : نقطتين فوق بعض و نقطتين فوق بعض تانى .. زي كده :

double Complex::GetReal()

Return type
of member
function

Name of the
class

Name of the member
function

{
THIS OBJECT real part بترجع ال design بتاع ال // وجوها نكتب المطلوب منها تعمله . فدى زى ما عرفناها فى ال function implementation خلص
return real;
}

كده اول function implementation خلص
ـ الجملة دى بس؟؟ 0.0
ـ اومال انت فكرك ايه D:

طب هى جابت ال real ده منين؟
ما فى ال main بقى هنبقى مدulinin ال complex number وكل حاجة و بعدها ننادى على ال function قفمام يعني .. كمل
بس معايا للاخر و الدنيا هتووضح
ـ خ بالك هنا الدكتور قال اللي بيكتب بدل ال cout<<real
ـ بينقص عشان احنا مش عايزين نعرض لل user بال main احنا هنا هنرجع لل user بال real اللي لسة هنستخدمه فى حسابات
ـ نواتجها اه نبقى نعرضها لل user فى فرق

نكتب كمان function

void Complex::SetReal(double x){
real=x;
}

ـ لو متاخبط بص على ال design بتاعهم هتقهم.
ـ اللي بعده

bool Complex::ReadComplex(){
Cout<<"enter real part";
Cin>>real;

```

Cout<<"enter Imaginary part ";
Cin>>imag;
return true;
}

```

اشمعنى هنا في Cin و Cout ؟

علشان لو رجعت ل function design دى المحاضرة اللي فانت ... استى هجبيهولك D :

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

فاحنا بنأخذ من ال i/p standard مش زى الباقي بتبقى حاجات من ال THIS OBJECT اللي بنبقى مدخلينه من قبل كده . منبنباش محتاجين ال i/p standard .

يمناسبة ال function اللي فانت دي بقى .. فيه exercise حلو الدكتور قال نمشي ايدينا فيه و هنتعلم منه حاجات كتير ال exercise ان احنا نعمل نفس ال function اللي فوق دي بس نقرأ الرقم كل على بعضه ك string من ال user (حاجة زي كدة : 12.345 -j5.213) .. وانا بقى اطلع منه ال real & imaginary ببني myself وقال في محاضرة المجموعة الثانية ان لو حد عمله صح ووراه للدكتور ممكن بيقى له بونص .. بس مقالش كدة في المجموعة الاولى فيعني الله اعلم

.. المهم يعني نكمel .. نكتب كام function كمان :

```
void Complex:: Add(Complex c) {
```

```
real = real + c.GetReal();           // or we can write so: real = real + c.real;
}
```

ثواني بقى كدة واحدة واحدة نشووف ايه اللي حصل .. مبدئياً Add دى المفروض انها هتزود ال real part بتاع instance من نفس ال class بتاعنا اللي اسمه Complex على ال real part بتاع ال THIS OBJECT بسهولة (اللي هو ال variable اللي اسمه real) .. طب فدلوتي احنا نقدر ن access real part بتاع ال object بتاع this member function call ال اللي اسمها GetReal .. او ناخده من بالنسبة لل real part بتاع c ؟؟ .. نقدر نجيده لما ن call هذا member variable اصلاً ؟؟
بس هو تكاثرت الأقوال حول هذه النقطة .. الناس ال restricted concept او ي ب encapsulation violation للكونسيبت دة .. بس هو ينفع لك syntax عادي بما إن ال يقول لا مينفعش تعمل c.real عشان ماتعملش violation عادي بما إن ال class اللي احنا شغالين بيه ((يعني لو كان ال class مختلف ماكناش هنقدر نعمل كدة))

```
double Complex:: GetMagnitude () {
```

```
double sq;    //define a local variable called "sq"
sq = real * real + imag * imag;
return (sqrt(sq));
}
```

طيب احنا ليه بقى عملنا declaration ليه مدخلتوش في ال local variable بتاعت ال member variables زى real & imag ... الاجابة عشان دة مش من ال ADT بتعنى، يعني مش حاجة محظوظة جوة ال black box بتعنى، لأ دة حاجة local احتاجتها في وقت معين وبعد كدة رميتها ومحاجتهاش تاني فعشان كدة حطيناه local variable

كدة احنا خلصنا ال source file & header file كذا .. نكتب بقى ال client program اللي ال user هيكتبه بقى :

```

main.cpp
#include "Complex.h"
void main () {
Complex x,y,z;
double u = 12.345;
double v = -5.223;
x.SetReal(135.43);
x.SetImag(-15.4311);
y.ReadComplex();
y.Add(x);
z.SetReal(u);
z.SetImag(v);
z.multiply(y);
cout << "Real Value " << z.real;
cout << "Imag Value " << z.imag;
}

```

كدة تقريبا الدنيا وضحت فتعالى نجمع بسرعة اللي اتكلمنا فيه المحاضرتين اللي فاتوا .. في الاول ال designer كتب ال driver بيتاع Complex.h و Complex.cpp ، ومن خلال الفايلين دول عرف يعمل ال ADT بيتاعه بكل التفاصيل وال constraints اللي هو عايزها

انا بقى ك user عايز آخد ال ADT دة واستخدمه في ال application بيتاعي هكتب ال main code وفي الاول هعمل include لل Complex.h ... عملت instances من ال class اللي اسمه Complex وعملت variables عاديه ومسكت ال instances اللي عملتها دي وعملت عليها operations مختلفة (من غير ما احتاج اعرف ال SetReal دي من جوة اعمالة ازاي او معمولة ازاي) .. بس انا جيت استعبيت في السطرين اللي بالاحمر دول وقلت هـ access على instances ... بس لما جيت عملت compile error طلعلـi private members لأن انا المفروض ماليش على compiler فال real & imag مش عارف ايه اللي انا كاتبه دة .. وبكدة ال designer فعلاً عرف يعمل black box ماقدرش انا ك user اي حاجة فيها غير عن طريق operations معينة وبس

يعني في الآخر انا عندي 3 files دلوقتي:



انا دلوقتي عايز بقى أ integrate ال 3 فايلات دول مع بعض واعمل run للا بلاكيشن بيتاعي ... ممكن اشتغل ب IDE زي eclipse وادوس على run وهو هي عمل كل حاجة او ممكن اعمل انا الكلام دة بنفسي .. اكتب الكود بيتاعي على اي text editor زي notepad مثلًا وبعد كدة المفروض الكود بيدخل على كذا stage بس الدكتور اتكلم عن اللي يهمنا بس اللي هو ال linker stage compile stage وال Complex.o (object file) و هيدخل له compiler هياخد الفايلات ال .cpp بس .. هيدخل له Complex.cpp فيطلع main.o (object file) .. المهم انه بيطلع main.o و هيطبع main.obj او في بعض ال machines خلي بالاك ال .h مش بيدخل في ال compilation عشان هو مش اكتر من فايل فيه ال declarations و توصيف للحاجات اللي ال user هيستخدمها بعد كدة .. يعني هو بيتاع ال user بمعنى اصح)

بعد كدة ال object files هتدخل على ال linker .. ال بتاع دة له كذا وظيفة بس اللي اتكلمنا عنه في المحاضرة انه بيـ link ال اللي موجودة في كل الفايلات دي مع بعضها .. وبما اتنا اخذنا ال physical & virtual address الترم اللي functions

فات ففيه معلومة زيادة ان ال linker هو اللي بيـ link ال physical & virtual addresses مع بعض .. بعد كدة بقى هيطلع ال executable file اللي هو exe .. او لو شغال بلينكس هبيـ out.

شوية كلام كدة بقى الدكتور اتكلم فيه في اخر المحاضرة ..
لو انا مش عايز ال user يشوف ال source code بتاعي (اللي هو Complex.cpp) .. عايز اخفي عنه الكود دة خالص وبيـقى شغال على عماه .. فممكن اديـله ال object file على طول فكـدة ال user لما يفتحـه هيشوف واحدـ واصفار فانا كدة خبـيت ال source code عنه ... بس انا لازم اديـله ال Complex.h عشان يعملـه include وعشان بقى عارف يستخدم ال functions دي ازاي ... وبالمنسبة المفروض عشان ال user يعرف يستخدم ال functions صح بقى الدـيزـاينـر عـامل return pre & post conditions كـدة بيـقولـ فيه ال documentation بتاعـ كل function وال documentation type (الكلام اللي عملـناه المحاضرة دي اللي فـانتـ) بحيثـ انـ ال user بـيقـى فـاهـمـ كـوـيسـ هو بـيـتعـاملـ معـ ايـهـ

سؤال مهم بـقـى .. هلـ الـ C++ـ بيـدعمـ الـ encapsulationـ fullـ؟؟ .. يعنيـ هلـ فـعلاـ الـ userـ مشـ بيـقدرـ يـشـوفـ الليـ جـوـةـ الـ black~boxـ؟؟ـ الإـجـابةـ لاـ .. الـ userـ لوـ فـتحـ Complex.hـ هـيلـاقـيـ انـ فـيهـ حاجةـ اسمـهاـ realـ وـحـاجـةـ اسمـهاـ imagـ وـدـولـ حاجـاتـ privateـ وجـوـةـ الـ classـ الليـ اسمـهـ Complexـ .. فهوـ كـدةـ شـايفـ أنـواعـ الدـاتـاـ الليـ مشـ عـاـيزـينـهـ يـشـوفـهـاـ .. اـهـ ماـيـقدـرـشـ بـ accessـ الـ الدـاتـاـ ديـ غـيرـ بـالـ setـ والـ getـ وبـالـ شـروـطـ والـ constraintsـ الليـ اـنـاـ كـدـيزـاـينـ حـاطـطـهـاـ بـسـ هوـ stillـ شـاـيفـهـاـ، فـماـنـقـدرـشـ نـقـولـ اـنـاـ كـدةـ عـامـلـيـنـ full~encapsulationـ .. اـحـناـ كـدةـ عـامـلـيـنـ partial~encapsulationـ

طبـ هـلـ نـقـدرـ نـعـملـ full~encapsulationـ بـ C++ـ؟؟ .. اـهـ نـقـدرـ بـسـ دـةـ عنـ طـرـيـقـ اـنـنـاـ نـعـملـ حاجـةـ اسمـهاـ interfaceـ .. بـنـعـملـ classـ وـهـوـ دـةـ الليـ الـ userـ بـيـشـوفـهـ وجـوـةـ الـ parent~classـ فيـهـ child~classـ فيـهـ

فيـهـ الـ private~membersـ ديـ حاجـاتـ لـيـهاـ عـلـاقـةـ بـالـ inheritanceـ الليـ مشـ هـنـاخـدـهـ فيـ المـنهـجـ بـسـ مـمـكـنـ تـسـيرـشـ فيـهـ

لوـ عـاـيزـ

المهمـ انـ اـحـناـ كـدةـ خـلـصـنـاـ المـحـاضـرـةـ ٨٨