

first lecture of object oriented programming using C++

C++ هي لغة برمجة أكثر تقدماً، تم إنشاؤها من C كلّاً من C و C++ مختلف programming paradigm (paradigm) حاها والـ C++ programming paradigm هي طريقة كتابة الكود أو طريقة حفظه من خلايا كتابة الكود من اختلافها.

- ١- C++ هي طريقة الأقدم وكانت خطيرة، غير مستخدمة (old)
 - ٢- C++ هي طريقة كتابة البرامج تعدد على الإجراءات (functions) من الـ C
 - ٣- C++ هي طريقة كتابة الكود أو طريقة حفظه خلايا كتابة الكود من اختلافها.
- ⇒ C is a procedure programming language (١)

ـ٤- C++ ← OOP مماثل مع C++ ← تتعامل مع data على أنها كائنات ميزة ← object oriented programming

Procedure	Procedure paradigm (paradigm)	Procedure Liner	Procedure Liner
ـ١- عبودية	ـ١- هي أفضل من لا Liner و productive paradigms	ـ١- الطريقة الأقدم و المستخدمة	ـ١- machine language
ـ٢- في حالة وجود الكثير من الرسوميات functions في برنامج واحد وكل مجموعة من functions تجري على data على مجموعة من data لا يتم وضعها في data	ـ٢- تتحقق جميع الفوائد	ـ٢- وصفها صعب و غير مستخدمة	ـ٢- غير عبودية
ـ٣- global Variables	ـ٣- Readable	ـ٣- تكتب بشكل أو لذلك	ـ٣- صعبتها في القراءة
ـ٤- هنا تحدث مشكلة لقراءة الكود فلا يوجده ما يحرك إلا مجموعة الـ globals و دخل مع functions	ـ٤- reusable	ـ٤- يتوجب على قارئه كتابة الكود من scratch	ـ٤- not reusable
ـ٥- Readable	ـ٥- maintainable	ـ٥- في حالة صعوبة تصميمها غير متوازن من الـ Liner	ـ٥- not maintainable
ـ٦- عن إتاحة data لطامح حفظها أنني أحياناً	ـ٦- ووجود functions مثل كتابة الكود كثيراً و صعوبة التغير من الـ Liner	ـ٦- لا يحصل على المرونة	ـ٦- not extensible
ـ٧- تغير فيها معاييرها	ـ٧- procedure	ـ٧- بالإضافة إلى الكود تكون صعب	
ـ٨- من حيث المرونة	ـ٨- procedural	ـ٨- من حيث الـ Liner ليست كذلك	

يتطلب التعلم على هذه المفاهيم بوساطة Paradigm جريء وهو OOP ومثال على ذلك Java, php, C++, C#، ...

- تلخيص لعموريه القراءة
- الـ function = المترددة فقط فهو يكتفى أن تـ data لا Access فقط وتغيرها.
- في حالة تم إثبات الـ data من function، خارجية ← الـ function → المسؤول عن الـ data
- وهو الذي يتوصى الـ data ليعلم دوافعه أن مفيش function خارجي يقدر يعدل أو يغير ولا data مش بتاعته ← مستشاراته بدوره متغير في الـ data.

إذا اللغة لا C++ هي نسخة محددة لغة لا C نفسها

Super C++

Able Sub C-

Pillars of oop

المبادئ أو المركائز التي تقوم عليها oop

- 1- Encapsulation
- 2- Abstraction
- 3- Inheritance
- 4- Polymorphism

ملاحظات ورسائل من كتبه للـ C++

- 1- الـ Extension بناء الـ file هيئته .cpp.
- 2- الـ Standards فقط تحوالى بعده اعتماد الـ C++ Compilation من لغة الـ C++.
- 3- الأفضل تطبيق المتغيرات جميعها أولًا قبل البدء من أي operation.
- 4- الـ keyword declaration معرفت من أي مكان كان عنوان الـ C++ يتخلص منه فيه

#include <iostream>

5- في حالة الـ struct مشتغل لازم أعمل إفاده لـ data لـ data على عنوانها

6- للطباعة نستخدم printf = cout <<

scanf = cin >>

لرقة كتاب

Cin, Cout هنال

Cout << "ID: " << ID;

Cin >> ID;

(%c & %s (%d) ← Format Specifier

malloc = new

جزء مساحة في dynamic memory allocation

نستخدم new و بعدها اسم ال struct للحصول

على dynamic deallocate المخصص من dynamic free بذلك من delete

فما هي كتابة الا data access ب باستخدام

1- ① في حالة static memory allocation

dynamic ~ ~ ↗

فما هي C++ بـ access بـ استخدام ① فقط بعض التطرى هل اهنا ندخل

Call by value

Call by reference

ول

Reference

اهنا بنوصل لا data بـ reference طرق

int x = 10, ← Value - 1

بواسطة الا pointer بـ يشير على عنوان الا X

* Address of x = &x;
Pointer to Address

Address

2- عن طريق الا Reference هو طريقة مختلفة أكثر سهولة من الا Pointer

& Alias of x = x;
Value = x ← عنوان x

- تقطفال Reference يتوضع في استخدامها للأ Function او function
او Main مثل لذى التي في الاقواس غير الا Variable التي عرفته فقط (x) من الـ

- تحديد أنا بقول Call by reference ولا Call by Value بيان في Scope او function لفتشها

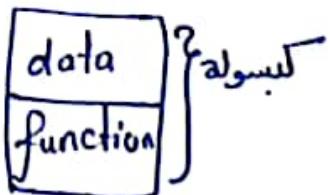
- في القوسي جنبها لو كتبت الا parameter يعني علامات = Value

- لو كتبت الا parameter ومعاه علامه = Reference

وهي التي أنا بقول ① في الـ function

• المبدأ الأول بعد مبادئ oop

في المحاضرة الأولى سرنا نبذة عن **Encapsulation** وعرفنا إنها تجمع **data** بـ **functions** في مجموعتين من الـ **functions** التي تستعمل لـ **data** التي تديرها وتحفظها داخل كبسولة



أ- جزء مخفي بخط فيه الـ **(private) data**
ب- جزء مطلوب بخط فيه الـ **(public) functions**

لهم يعلم

١- يحسن الـ **data** من نفس وصوله إلى عتاده لو كانت **global** وأنه قد يغير يومها ويفترى
فيما يحصل مشكلات كثيرة

٢- محمد يقدر يوصل غيري فار مستقبل البرنام اهتمام أزود عليه **Feature** جديدة حسبيه أسلوب
٣- لو حصلت مشكلة وأهتجت لحلها أو أحتاج إلى التدخل في الكود دايرد ومن ثم هيسعى
أمثاله تانية ويسهل لها

٤- قراءة الكود هستلوي وأسهل لونه تاني أشتعل على نفس البرنامج لأن الـ **capsule** لم يجتمع
data فيها على رقم بعضها **functions** بتأثيرها
يعنى بهذه الرؤية **Encapsulation** قدرت تحقق **productivity** (Extensible - Maintainable - Usable - Readable)

• ملحوظ تأخير البرنامج في كل مرة

-
١- **Data Validation** → (٢٤)

٢- **effect of change**

وهل ده ممتاز ! الـ **capsule** معزولة عن بقية البرنامج ولا يؤثر على

هنا ينضر بعد الجزر الـ **public** في مثل سبائك **→** يوصل بين الكبسولة وبقية البرنامج
لكن الميزة هنا أنه محمد من الـ **functions** للـ **data** الأسئلة يقدر داير
الـ **data** السوجه

٣- **data access** ← المسؤول الوصي على الـ **data** **public functions**



Access modifiers \leftarrow (Public, Private)

يحيى الا Modifiers الوصيارة القارئ على الوصول للـ data و تؤثر على مرويـاـ ما يدخل الـ كبسولة

٢- الـ Functions لـ تفعيله

• نوع القديم الـ stand alone

• نوع الجديد الـ member مع الـ كبسولة

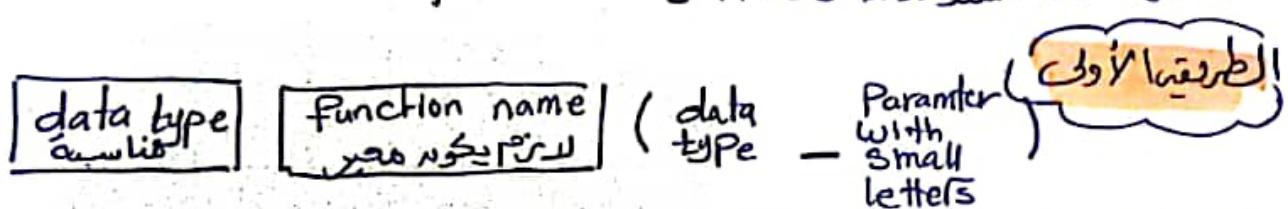
الأفضل استـ رام الـ stand alone
يمكن اسـ تـ رام الـ stand alone
وجود عـ الـ stand alone
بيانـ بـ الـ stand alone

stand alone	member
<ul style="list-style-type: none"> - يتم تعريفـه في الـ global data - غير قادر على الوصولـاـ data بشكلـ مباشر <p>Parameters</p> <p> تحتاج من Parameters لا تحتاج من Parameters</p>	<ul style="list-style-type: none"> - يتم تعريفـها داخلـ الـ scope الكـ بـ سـ وـ لـ - قادر على الوصولـاـ data بشكلـ مباشر

٣- طـرـيقـة كتابـة member :- يوجد طـرـيقـتان

١- اـنـوـدـ اـنـاء إـنـشـاء member داخلـ الـ كـ بـ سـ وـ لـ

٢- اـلـثـانـيـة اـنـاء اـسـتـعـادـ الـ function في ايـ اـنـاء



(member function) \rightarrow (مثال على تـعـيـدـ الـ function bool withdraw(float amount))

(int, float, string, bool, void) Data type \leftarrow

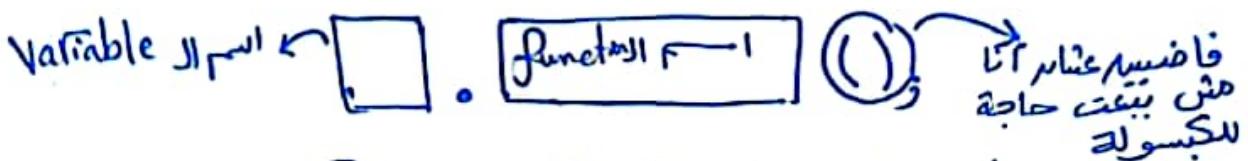
\leftarrow لـ زـمـ نـخـتـارـ الـ function عـ حـسـبـ الـ Data type يـ جـلـ إـلـيـهـ بـعـدـ لـوـهـ جـلـ وـظـيـفـتهاـ
إـنـاـ تـطـبـعـ فـقـطـ هـسـتـخدـمـ (void) لـ إـذـامـشـ هـتـرجـعـنـ قـيـمةـ إـحـاـلـوـ هـجـلـ وـظـيـفـتهاـ
نـقـلـ عـ عـرـهـنـلـ هـسـتـخدـمـ (bool) لـ إـذـهـ أـمـطـ بـرـجـلـيـ Checkـ

الْبَرَاءَ اللَّهُ بِهِمُ الْأَقْوَامُ ۚ

اوند لازم يحتوى على حاجيسيه: **datatype** مناسب **parameter** من المطلوبه نفسها هى من برا

- جرى العرف في كناب، او **parameter** يجري Small وقبلها لازم (→) .
← **ل** كيّت حواها وظيفة الـ **function**.

المرفقه الثانية، عند استدعاء الا Member برا الائمه



كما في Compiler هو الذي يعدل المقدمة في سطر برمجي إنه ينسخ عنوان الـ `this` إلى `member function` التي تادي على المقدمة `function` بالإضافة إلى كل المقدمة `this` التي تادي على الـ `Function` ويعطيها فوهر من الـ `function` بالإضافة إلى كل المقدمة `this` التي تادي على المقدمة `member function`.



this is a pointer to the caller object
هذا هو مُنَصِّبٌ إلى المُخْلِصِ

معلوماتی الگاری خریده عمل Dynamic memory allocation C++ می باشد

اوّلین لائن مِنْ اكْوَهْ عَالِمِ أُمَّةِ الْجَاهِلِيَّةِ يَسِّيْلُ الْمَحَالِ بِالْعَدْوَنِيَّةِ

employee * A = new *employee*;

☞ فقرة جزء مساحة بالطابع خالا heap متوقفة على ذيكر pointer لارتكابه لذاته هنـى بـتـبعـتـ فـطـلـيـرـ لـلـ heapـ لـتـ بـتـبعـتـ عـنـادـيـهـ فـقـلـ وـتـشـارـرـ عـلـيـهـ باـسـتـخدـامـ لـr pointer

Encapsulation

lecture 3

و مدخلات الـ
Polymorphism

⇒ في المخاضرة الثانية عرفنا أذهبية encapsulation و inheritance، و بذوق الـ polymorphism التي يتساهم بها oop، ولكن في بعض الحالات تكون تحصل مشكلة

هناك لوناً علينا أن نرى الأكونت في بنك و محتاجة أحاط قيمة إيداعية للأكونت
ده [المعلومة دى هي معرفة كل شوية قد يغيرها كل بكتبة من واسطة فلقم] يعني معرفة
العمل يكون حاطط قيمة إيداعية للأكونت بـ ١٠٠ جنيه و على بعد فترة لا تخزن توكودين
المشكلة دى يحصل بسبب إن function المسؤولة عن إنما تضع قيمة إيداعية هي
(SetBalance) public function يمكن استعادتها بسهولة ووضح قيم جريدة حقول (SetBalance)
⇒ الكل هو إن أعمل special function ← بتنادي مرة واحدة فقط اثناء ما يجري
الأكونت بطبع فيها القيمة الجديدة إيداعية اللي عاوزها وقدرتني أنا ربطت

Special Function ← Constructor

- بنادى على طرق واحد فقط اثناء ما يجري object
- المسؤول عنها بعد كده هو الـ Compiler لذلك يجب تعيين اسم مميز وهو نفس اسم داد
- مني برجع حاجة ومني بتاخد datatype

Account () {

```
    this → Number = 1;
    this → balance = 1000; }
```

Pointer to the caller
object

أنا كده فربت أعمل لـ اثناء عاوزه إن أحاط قيمة إيداعية للـ balance بـ ١٠٠٠ ما أقدر أقدر
تاف، ولكن بدوره قصد مني أنا ثبت رقم الحساب عند الرقم (١)

هل دى مشكلة ؟ أيعنى طبعاً

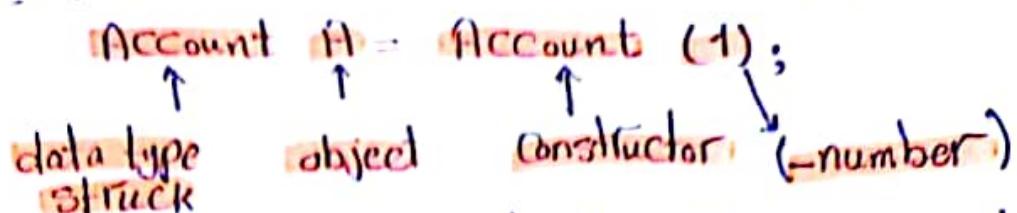
لده لو عندي ١٠٠ علامات متراكمة مسارات على أرقام حساباتهم كلهم هنطبع بـ ١٠٠٠ غلط
number المشكلة دى لأن لما أعمل الـ constructor أعملي على أساس يتبعنيطاً قيمة الـ number

هيبيجت حال الـ parameter المسمى (-number) (number)

```
{ this → (-number),
  this → balance = 1000; }
```

إذاً بعثت الرقم الى المبني على main (number) parameter

➁ بناء على المبني على main دعوة المبني الى constructor (number)

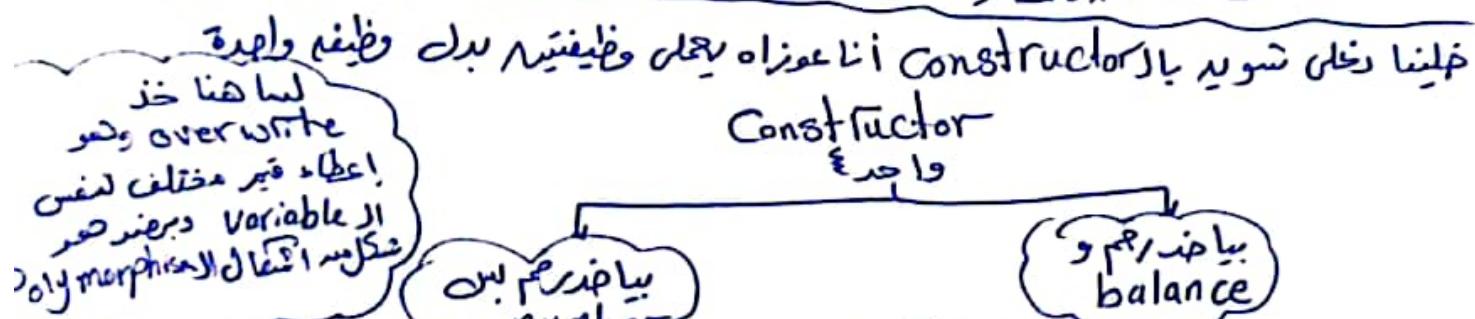


الخطوة مبنية ان بعثت object اسميه Account(data type) الى اسم نفس الـ constructor (calling constructor) number .

ويمكن قدرت احاطة Encapsulation و حفظ قيمة ابتدائية لا يغيرها.

ولنفس الكيفية لو عايزه اخدر قيمة الى balance مكتبهش انا hard coded balance الى Constructor قيمة ابتدائية للbalance اداً الطلاق تتجهن فـ balance واحد تستقبل فيه قيمة number واحد تستقبل فيه الـ amount

Account (-number, -amount)



خلينا نخل شوية بار Constructors اتعزاه بعده وظيفته بدل وظيفه واحدة لسا هنا خذ overwrite او Function Overloading بعدها غير مختلف نفس الـ Variable ويعنى صور شكله افعال المorphism واحد

الـ overload دعوة لـ function باسم واحد تحمل التردد

والـ overload هو شكل من اشكال الـ polymorphism بمعنى تعدد المؤجرين . If polymorphism اعم و اشمل من الـ overload .

الـ biject اني بنادي على المبني على function اسـ. function و لكنه في المبني على عدد وظائف مختلف

إذاً الـ polymorphism يحتوى على productivity .

• Readable : قراءة الدخول سهل لغله عدد الأقواعد لاف يستخدم كلام واضح

• Reusable : يعني بعد استخدامها لكن بطريقة efficient

• Extensible : قادر ان أضيف feature جديدة

• maintainability : التغيير سهل

قبل ما أدخل في مشكلة maintainability المثلث المسككه حتى من فيه
 في لغة C++ أقدر أعرف ألم منته functors

parameters كلام لهم لهم (اسم واحد ولكن عدد مختلف)
 مثال

Sum(A)
 Sum(A, b)
 Sum(A, b, C)

العلاقة بين اسمها binding ← ربط

أنا عندي ألم منته function بنفس الاسم هو موجود بينهم يربطه بين الأفواه ←
 Parameters ←

2 functions لهم نفس الاسم

الـ Compiler يقدر يفرق إزاي؟ بين ←
 Signature ← ماهو الـ Signature عن طريق الـ

1- نوع الـ parameter
 2- عدد الـ parameters
 3- ترتيب الـ parameters

(int x, float) ← parameters

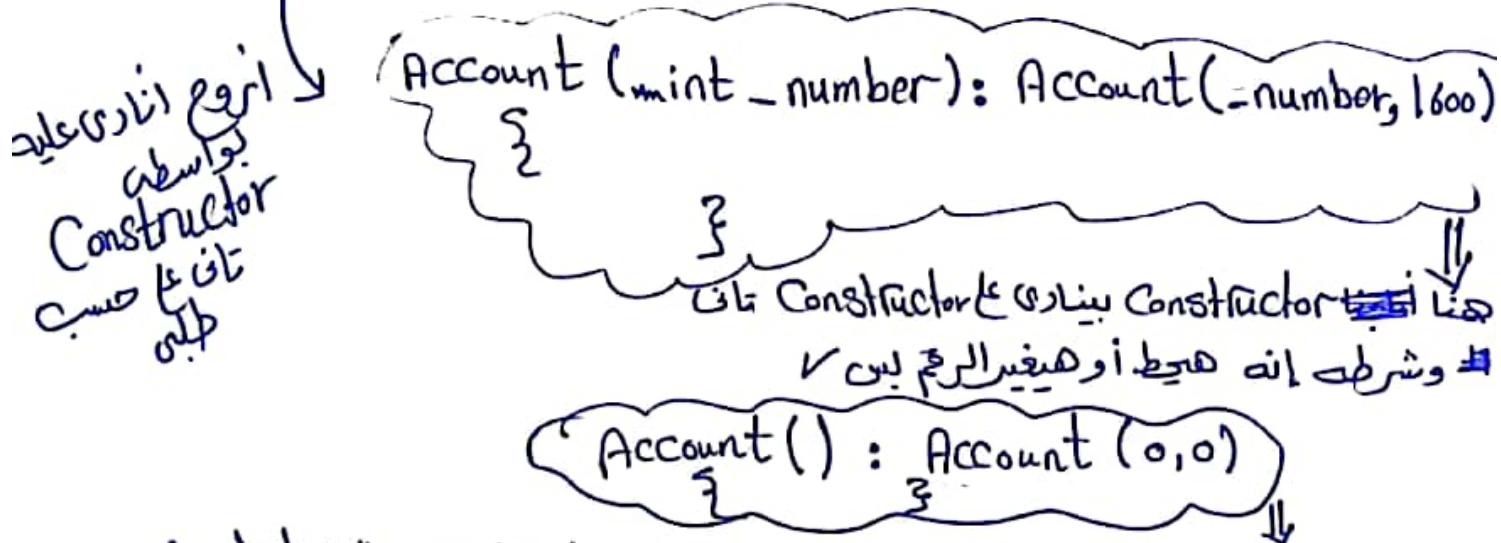
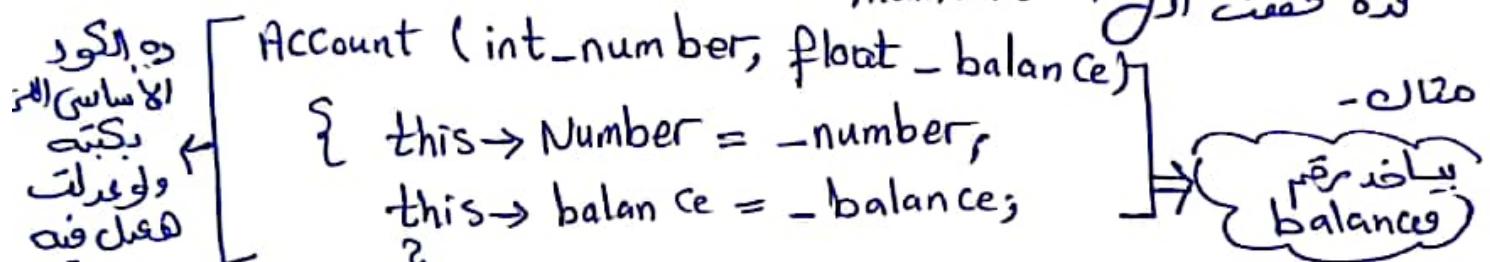
وبيكِه حققت
لـ Readable \rightarrow كتابِ الكود مختصر ومبصر وسهلة \leftarrow
لـ Reusable \rightarrow يستخدمها كثير باسماح مختلف
لـ Maintainable \rightarrow فرجع لمسكك له

lecture 9

المشكلة في maintainability هو لوعنة تعديل او تطبيق واحد ورثته مهني على كل الـ **functions** التي متصلة بنفس الـ **class** فكلها تتعرض لأحد من أكثر من مكان **maintainability**.

لو وصلت للخطوة دي ضاع المـ **constructors** فيه **Constructor** فيه كل الشغل كله وأثارت عليه لو اسفله **Constructors** الثانية دي خط تعلم يلارتا بالشكل اللي تجده تانية وار **Constructors**.

اذا \rightarrow كاتب الكود هكبة مرة واحدة ولو جست أدخل تعديل في مكان واحد عايزه أضيف **feature** جديدة تسرى على الكل برهنو هكبيوا من مكان واحد فقط كده حققت **maintainability**.



هنا **Constructor** يبني على **Constructor** يعني **Constructor** يبني على **Constructor** وشرطه انه هحيط او هغير الرقم بين ✓

لوعاظه أدخل تعديل لسرى على الكل بعد من أول **Constructor**.

لوعاظه أدخل تعديل خاص بالمثال ٢) **Constructor** ينبع من الأسلوب **maximum maintainability** ومنه نصيحة بدل ويجده قرر ان كل الـ **Constructors** ينتهي على الـ **Calling** **Constructor Chaining**.

على الـ **Calling** اللي تمت بعد الـ **Constructors** سلسلا.

Constructor Chaining ← تسلسل.

التشعیب main element في oop حوالء ولهم default parameters.

الكتاب \leftarrow **defualt Parameter** هو **Parameter** عادي حيث ولكن بطله قيمة ائتمان الكتاب
حسبما يرى المبرمج فاينما تغير قيمة **Parameter** في المجموعة التي تم ادخالها

Account (int - number = 0, float - balance = 1000)

وَاللَّهُ هُوَ مِنْعَلِيشْ فَيْمَا نَاهِطُ الْقِيمَدِيِّ مِنْ نَفْسِي
by default

default من شروط الا

- $(x=30, y=20, z=10)$ parameters لایه ای از قم

Account(0,0) \leftarrow 0 (Initial
Account)
Account(number,100) \leftarrow Initial & 100

لأننا نتذكرة إلى مشـعـارـه أـعـدـهـ عـلـىـ الـF~unctionsـ مـسـقـيـلاـ فـالـأـقـلـ إـنـهـ يـكـبـيـوـ
Codeـ الـS~tat~icـ وـخـصـصـهـمـ مـنـ الـC~on~str~uct~orـ d~ef~ault~p~ar~am~erـ

Lecture 11

Static data
Destructor
friend function

Compiler will call constructor if there is no static data or friend function in object. If there is both, it will call destructor first and then static data or friend function.

default parameterless constructor \Rightarrow constructor if no parameters
 مجرد ما أعمل Constructor لا يوجد Constructor مع параметر
 صورة المثلث المفتوح يشير إلى أن Constructor لا يأخذ أي معلمات

Access modifiers
struct
public by default
VS
class
private by default
public by default
Access modifiers
ولكن في الحالات التي لا تأثر فيها المعلمات على المقدمة، يمكن استخدام المعلمات العامة (public) أو المعلمات الخاصة (private) بدلاً من المعلمات الموصولة (access modifiers).

Instance data وتحتاج لوجود معلومة مشتركة بين جميع الـ Instance وتحتاج لـ Static data

C. Static data

ما هي الـ static data ؟
Shared data \Leftrightarrow Instance data \Leftrightarrow data segment (موجودة في كل Instance)

static int count; , static string Branch;

لدينا اسفل الـ data type لا يغير الـ value له

scope resolution operator :: العلامات الدالة اسمها
by using the static data type يتغير الـ scope resolution operator

Account brow! data type متغير

Account :: Count = 1; قيمة

ما هي الميزة هنا؟ الكود ده يكتب قبل الـ main

وليس تابع لـ scope وليس تابع لـ static data

براعي كتابة قيم نوع الـ data type المناسب مع الـ static data

int Account :: count=0; / String Account :: Branch = "Aswan";

lecture 4

للسير الكود ينفيه، $\text{int Account::Count} = 0;$

- ما يكتب static data أنا بجزء من حياة البرنامج كله وارد جزء مستخدم
- الـ static data أصلًا خارج Compiler لأن احتياجاته من بغير من عليها أكتب الكود
- لوحدة قبل الـ main وارد للـ static data قيمة وبعده الكود يعتمد على Compiler
- الـ 4Bytes أنا بجزءهم دول بالفعل مستخدموهم
- وارد يكون التقى هو غير صحيح ولكن الكود هو لو تكتبني الا static data من الممكن
- لو أنا عايزه الـ User هو اللي يطأقيه أو static data إذا لازم أعمل Count تأخذ منه الـ User الفيقيه اللي هيطلها ويورثها تساوى بـ Count وبيان الـ static data كان هتكون function

ما هى الـ static data function \leftarrow static function

طريقاً أن تعامل مع الـ Instance data

`static void setCount(-count)`

طريقة كتابة
في الـ struct

data types بنادي عليهم باستخدام الـ static
ما ينفعنا static و مكتوبة تحت السخة
يعنى لها

قد تكون أكست كده
A.setCount
أو A. يعنيها

طريقة استعمالها (قيمة)
في الـ main

مقاييس static data

Instance data	Static data
<ul style="list-style-type: none"> data تغير نسخة يعنيها التغيير فيها لا يغير في قيمة الـ instance 	<ul style="list-style-type: none"> Instance data shared data يعنط ولكن التغيير فيها يظهر عند جميع الـ instance
<ul style="list-style-type: none"> الـ instance function A.Display(); مثال A.setNumber(); 	<ul style="list-style-type: none"> Instance, static functions يمكن للـ static function عليها.
<ul style="list-style-type: none"> لا يجيء الـ static function 	<ul style="list-style-type: none"> لازم نكتب الكود الذي يعني أن تغير الـ static

~~int Account::Count = 0;~~

~~string Account::Branch = "ASW";~~

Instance

Lecture 4

• هذه القيمة التي تعود من إنار function ~~عند~~ هكذا تستعمل أو Static data

* هناك حسابات البنوك التي تستخدم في شرطه ، إن لم يعنى معرفة عارضة كل ال Instance تسوفها ، وإنما static كده هي مشكلة لما هاتكون ثابتة عن الأقل وده بيتفقش في حالة حسابات البنوك ، ناك ما هتربيت حساب عاوزه أتزود رقمه عن المدحفات بواحد

لهمكنش سمح ! بـ اسْتَغْفِرُ لِكَ اعمل كـ static data

class Instance function → Job → static data creation

Number ← Instance data of class i

destructor

۱۱۴  هو مکان بخوبی Compiler کار object می‌بیند و destructor را ایجاد می‌کند تا در خروج از دستور بازگشت (return) این دستور اجرا شود.

طريقه المسميه يتبعه نفس طريقه الـ Constructor بالضبط لذلك يقع (~) [tilde symbol] قبل لا يفهم بـ Compiler (InTrainee) يعرف بـ Destructor

← object بہوت شکلیں مختلفیں ہے

- لوكار ال object مثلا stack هيومت مع نهاية الموس
- لوكار ال object من ال heap الذي انا هعمله فيها delete

Destructor	Constructor
<p>object یعنی معرفہ داری کے طور پر۔</p> <ul style="list-style-type: none"> - Cannot be overloaded - It can be polymorphic <p>Default Compiler provides default parameterless destructor by default.</p>	<p>object یعنی معرفہ داری کے طور پر۔</p> <ul style="list-style-type: none"> - Can be overloaded - Parameterless constructor by default.

إيه لارجعه الـ Destructor
لو عندك جموعه بيانات خاصه بـ object و اضفه في مساحة وعائمه أسماعها قبل ما يعوشه object
فنال \rightarrow نكون مخزنه أرقام ملائمه متربص في الـ heap يكتب منه كود الـ Destructor
dynamic memory \rightarrow Cleaning up unmanaged class
بعد آخر بندقته الـ

maintainability

Reusability ✓

Lecture 4

Lorentz function

2

private data || Accession function \leftarrow
و بالاتر وهي موجودة خارج الـ Struct || هي same datatype || int

friend void display();

طريق استعمالها ←

Encapsulation (بالرغم من ذلك يمكن للغير دخول المثلث) ← friendfunction

Lecture 5

Templates - Copy constructor
operator overloading

e

قبلها أدخل لموصن المحافظة النقاط وجب توسيعها.

ما هي هيكل البيانات Data structure

(٢) مجموعة من البيانات التي أنا عاوزة أجمعها أو أخزنها أو أرتديها بطريقه ما حكتني
من إجراء عمليات عليها أو التعديل فيها.

• كلّ من نوع (double, String, float, int) data type هو الّذي كاتب بسلاسل بتأليفه \Leftarrow Data structure.

ويمكننا كتابة أي خط معها يجوعه عن الـ functions المسئولة عن عمل التدريبات او اجراء عمليات عال data

← وعرفنا من المأمور أن نعاتب إتنا نعم في data structure بطيئين هما:

Structure: is a user defined data type

private by default < class .

public by default + struct

أنواع الـ data structures

--- String int) C primitive data type دالة ايجاد ختناها في الـ built in data structure بـ **built in** مكتوبة في المثلجة مع اللغة بـ **primitive** بـ **String** او **int** .

non liner

Lil Tree -

Labeled Graph -

میتوں لیے = non primitive

Liner

Congratulations Mr. Liner

فیفا میں ال C → Linked List

linked list
Eg: \leftarrow Stack

Lud \leftarrow Quelle

lecture 5

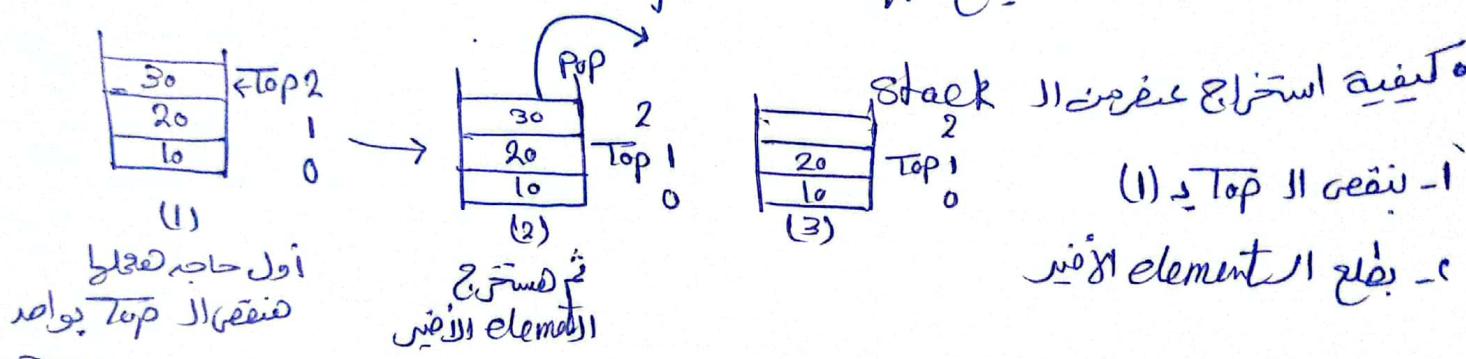
٣٢

يعرف متى انت الstack ناقص واحد $\Leftarrow \text{Top} \neq \text{Size}$
أي $\text{Top} = \text{Size} - 1$

لهم بعمل لك $(++\text{Top})$ يساوى عالي المكان الفاصل الى Top فيه
الاخير دخل هو اول واحد يطلع \Rightarrow element الجديد.

i.f.o Pop: return the top element of the stack

في عملية الاستخراج أنا يرجع آخر لـ element يعني بيسو الطريقة
آخر واحد دخل هو اول واحد يطلع \Rightarrow Last in first out



لهم جزاً نجزئ في طلاق او CheckPush

في طلاق او Push او Pop او CheckPush هل خاتمة stack ، لا او stack في
stack \Leftarrow عندي معلومة أقدر من خلالها أعرف إنما
فاضي ولا رث
1- \Leftarrow Top \Leftarrow إذا لو كان الـ Top يبدأ من العدد 0
stack \Leftarrow إذا لو كان الـ Top أفراده 0 إداً الـ stack

في حالة او Push أو CheckPush أتأكد إنما مكان خاتمة
stack \Leftarrow دخل فيه الـ element
عندي معلومته أقدر من خلالهم أعرف الـ stack
عالي ولذلك \Leftarrow size \Leftarrow معلوم بالنسبة
2- \Leftarrow Top \Leftarrow الـ elements \Leftarrow التي يشار على الأخر
 \Leftarrow Top = size - 1

1- \Leftarrow Top \Leftarrow if (Top > 0)
2- \Leftarrow cout << Item[Top] << endl;
3- \Leftarrow return Item[Top];

4- else {
5- \Leftarrow cout << "the stack is
empty" << endl;
6- \Leftarrow return T(); } \Leftarrow للرقم
7- \Leftarrow pop circled what type of
8- \Leftarrow cout << "Poped" << endl;

3 = size \Leftarrow stack \Leftarrow لو فضاً 3 يعني stack \Leftarrow void push (T item)
والرقم الـ Top \Leftarrow يكون أصغر من 3
لهم جزاً نجزئ في طلاق او CheckPush
1- \Leftarrow Top = size - 1
2- \Leftarrow cout << "Top = " << Top << endl;
3- \Leftarrow cout << "the stack is empty" << endl;
4- \Leftarrow cout << "Pushed" << endl;
5- \Leftarrow cout << "Poped" << endl;

lecture 5

٤٦

لو من الممكن المiscalفات حيث أن push (5,6) هي غير بار (5,6) عما يليه push
جديدة وهو خلل الـ size ينبع push كل مرة تغير push ل النوع push
مختلف push stack || overload push stack ده من أذى حل في البرئية دى

+ اعمل الأفضل هو إما عمل stack واحدة بس والا stack دى تستغل على أي نوع data
Templates وهم دى وظيفة الـ

الـ Templates هى خارج يمكن إعاده استدعاها من أخرى أو التي تدعى
template<typename T> طريقة كذا ستبطء

الـ T دى دلالة على data parameter أو الـ type parameter دلالة على نوع الـ data
(pass data type إذا أنت بعمل) قيساً على المفهوم overriding

polymorphism وهو شكل من أساليب الـ overriding

ماهو الـ overriding

هو نفس الشكل بنفس signature ولكن يقوم بأمور مختلفة (يكتلوا على نفس طبقته)
Template class .
Template function

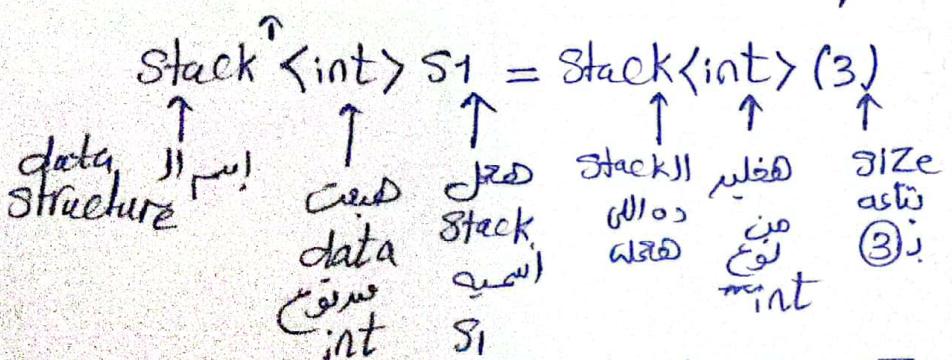
أمثلة الـ overloading نفس الوظيفة ولكن signature مختلف

Swap(int x, int y)

Swap(float x, float y)

stack of int

main من الـ pass datatype



فون من الـ structures كل T مكتوبه او Compiler

هي بيطلب وبيط مراكبها int

النسخة المنشورة للنسخة المنشورة Copy Constructor II

`Stack<int> S2 = S1;`

لعنوان الوكيل

S1 هي جملة من Compiler من نفس الشكل لنسخة جزءية الـ S2 بدلاً منه لفترة ١٥

يستخدم `default` also \leftarrow Copy constructor by default

ـ نسخ من عالي الوضوح \leftarrow Shallow Copying

نسخة جزئية

S2

لبياناته
محلها

Top	0
Size	5
Item	oof

نسخة

S1

Top	0
Size	5
Item	oof

لبياناته
محلها

المشكلة هنا في Top والـ Size والـ Item لـ S2 هي في التالى هي
ـ المثلث يشير إلىPointer
ـ الآنسنة يشير إلى مكان وآخر المعمورى

لـ S1 وـ S2 غيرت الفكرة بـ ⑦

ـ S1 وـ S2 غلط لـ دواف المفرضي 2stacks مختلفين

ـ إذا أكل هو إى أعلى **deep Copying** \leftarrow نفسه أو **Copy** القديم ولكن مع تعديل

ـ في جزء `Item` \leftarrow إذا أنا هعمل `CopyConstructor` override \leftarrow إذا عمل طريقة إنى أثبتت القديم من حيث `Top` وـ `Size` وأعدل فقط مثابع `Item` \leftarrow `CopyConstructor`

`+ stack (const stack & - old)`

ـ تثبيت لـ `Top` \leftarrow بـ `this`

ـ تثبيت لـ `Size` \leftarrow بـ `this`
ـ تثبيت لـ `Item` \leftarrow بـ `this`
ـ تثبيت لـ `data type` \leftarrow بـ `this`

ـ بـ `new` \leftarrow بـ `this`
ـ بـ `const` \leftarrow بـ `this`

ـ `this->Top = - old.top;`

ـ `this->Size = - old.Size;`

ـ `this->Item = new T [this->Size]`

ـ إيجاز المدخلات
ـ heap
ـ size

lecture 5

lecture 5

operator overloading

١١ operators نلات أنواع اهنا عارفين ما زوّجوا

• إيه الـ operator لـ overloading أعمل خليني أعمل

رمادن خالد C لاما تعرف variables وينديو فون كذا بقدر تستخدموه operators
لشكل هياش

int y = 20; (int x = 10;)

Count << X(+) << endl;

نحوه Compiler قادم او (+) ضغط لمزيد من تفاصيل الفكرة بخطيابه X ٦٧ و تجدهم

وكل من لغة C++ لها مفاهيم مختلفة data struct object بنوع امثال ما يلي

no 2 objects main عرفت main وجدت حفظاً Trainee نوع  (Trainee main) ↪ main

$$\begin{array}{l} \text{Trainee } X_i \\ \text{Trainee } Y_i \\ \hline \text{Delete } X_i \leftarrow X + Y_i \end{array}$$

class Trainee {
 int Id;
 String Name;
 float Salary;

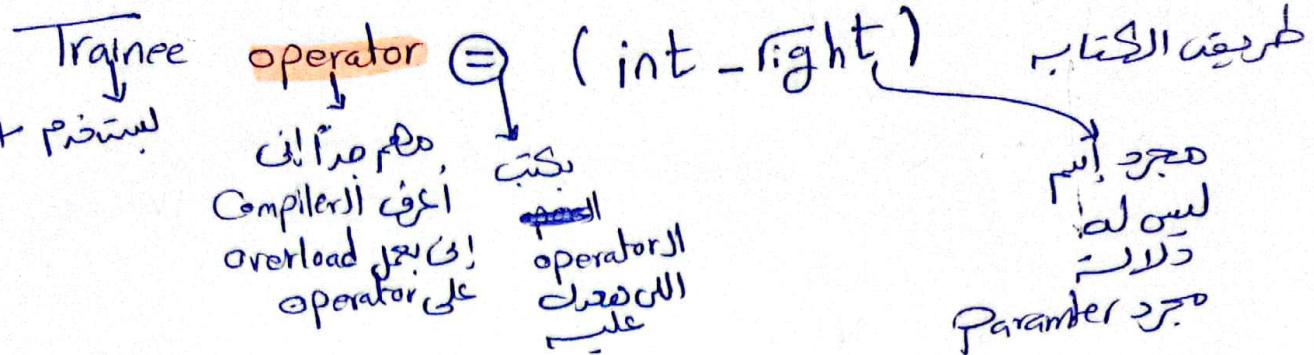
لهم حلة int الجح عدى من ملأ Trainee مدراش ؟

→ بالنظر لل Compilers فهم يأخذون علامة + بعده متغير ثم نوع int: درجات الملل
يتبعون (هنا كانت تألف data ونقول ~~objects~~ (اجعل objects من هم لهم الامر
. datatype بتحوى على الـ Trainee نوع

بعضیات کے اٹاگی حاجت اسی اگر Compiler نیکو اپنائی کا استعمال + objects میں features کے overload operator کو تعریف کیا جائے گا اسی کا انتہا

يبني ماعلاك بـ operators في class أو struct أو class يبني Compiler ولا classes

لأننا نعمل على operator المكتوب باللغة البرمجية \Rightarrow operator على



دفر الـ function يقول أنا قادره أعمل فيه

{ this \rightarrow Id = -right; }

يمكن الرؤم الى هيبيتك في الـ right - خط او ساوير بالـ Id او بعنه آخر لا ~~كذلك~~ الـ object من main تكون قادره الـ Id

Trainee T_i = 1;

هذا يدخل تلقائي الـ Compiler فيه لهم ان بعد الـ \Rightarrow لوجن int حده من الـ

قبل ما أعمل تعديل على الـ \Rightarrow حيث يمكنني ان اغير المسمى الـ Compiler كان هيزبلو object دو \Rightarrow هسيقول الـ T_i هو من 3 حتى انا اتدري قيمة 1.

over loading: two or more functions with the same name but different signature with similar functionalities.

لها نفس البرسم ونفس الوظيفة ولكن الاختلاف في signature

over riding: two or more functions with the same signature but different functionalities

لها نفس signature ولكن الوظيفة تختلف

over riding امثل على الـ

Templates
Copy constructor

overloading امثل على الـ

operator -
Constructor -
functions -

Lecture 6

Relationships between Inheritance

Association - Aggregation
Composition

مقدمة للمزيد قاتلها :-

private ← Attributes .
Public ← Methods .
... overriding ←
Copy constructor .
default parameterless constructor .

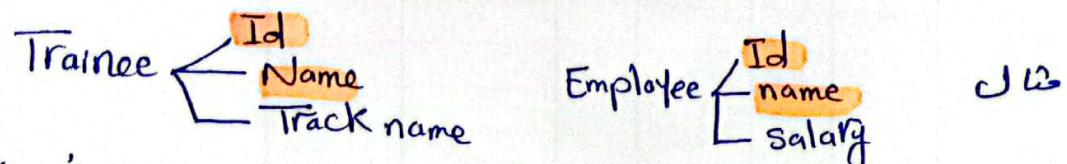
polymorphism (أمثلة كبيرة على ذلك) ← Templates .

Encapsulation ← Abstraction ←
• أصبح التعامل من خلال methods دون التعرض للـ data بشكل مباشر

Polymorphism ← Inheritance ←

الوراثة ← Inheritance

لوعندي 2 مترتبة متصلة من الذئاء ←
Class 1: Trainee
Class 2: Employee

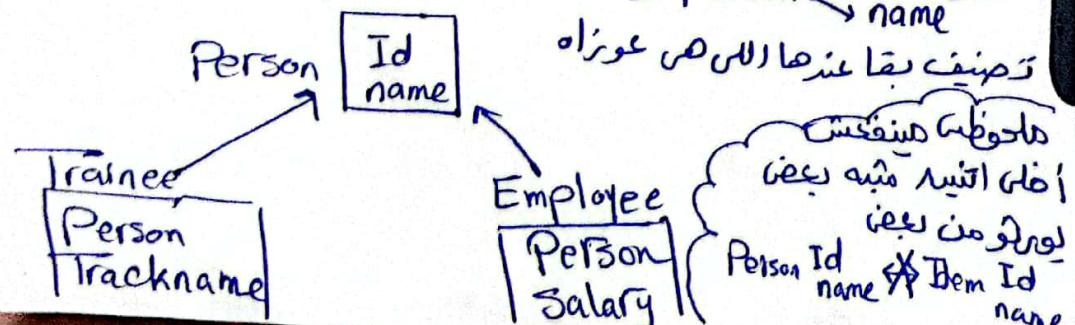


كلاس 2 مترتبة في التسلسل الأول هيقولك طيب ما أجمع كل الـ attributes من واحد

من واحد class ←
Case 1: Human ← الخطأ هنا إنك عارف إن عنده ما هتهيداره تكتب فقط attributes بداعياً هيلوودي واحد غير (Salary, Track, ...)

يعنى آخر أنا بيوظ الـ memory وستعمل هنا أجزاء من مستعاراً وهو غلط جداً -
كل الأقسام هو تجميع الـ attributes من الـ 2 classes ووضعهم في class صغير
مثال Person ← وآخر Person يورث من 2 classes

هو دا الـ Inheritance



Relationships between data types (memory - Lifetime)

عندما أقدر اشتغل على الـ Inheritance وأطلق الـ data types تعلم بعضها لازم أكون على دراية
بأنواع العلاقات بين الـ data types وتأثيرهم على حاجته ملخص :

١) **memory** : أنا من هسأل نفس الكود الله أنا عاملته هو هيستغر ولأذا أنا حصلت نفس
عمل سفله ده أفضل حاجة للـ **memory** لأنها ملحوظه .

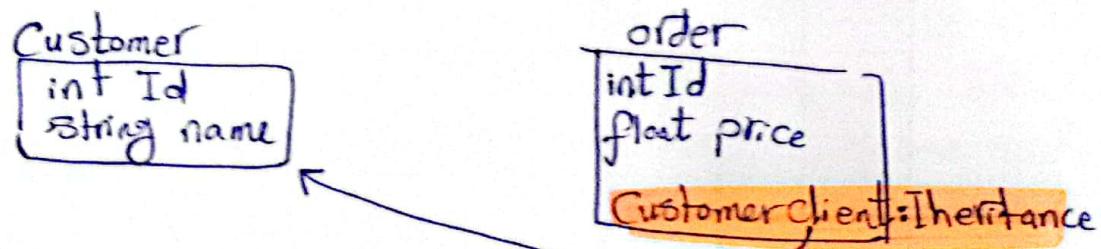
٢) **Lifetime** :- ثانية هسأل نفس هل موت أحد الـ data types يحيي موته الثاني ولا الثالث .

• طيب عساي أتحاكي عن المفهوم الأول ملزم أنواع العلاقات بين الـ data ألا إلخ :-

٤) **Association** + أضفت أنواع الترابط موب آخر دلالة data (يعني موطن المفهوم)
(Peer-to-peer) + هنالخ **Sense of ownership** ولكن بمعنى آخر **Aggregation**

٥) **Composition** + أنواع العلاقات وهذا لما المحتوى يعني عزمه الثاني يعني معاه لذمه حياته
الثانية ليس لها معنى بعد الأذول

لدينا برمجيات مطعم المطعم ده عنده 2 classes customer و order
customer (name , Id) 2Attributes هيكون عنده customer (name , Id) 2Attributes هيكون عنده order
order (Client , name , Id) 3Attributes هيكون عنده order
order (name , Id) هيكون عنده customer (name , Id) جواها تحمل client هن معلومات السلف التي اشتهرت وصنوبره نوع customer و customer



صلبه لو جينا نعمل أو public order أو public constructor اهنا عرفنا انه نعمل حاجة
لو كتبنا الـ prototype

order(int - id , float - price , Customer - client)

لو في المخطوطة على build قبل ما تكتب او function ده تجعل ايه او
No matching function for call to

Error و يقولك **هذا غير صالح**

لدينا حصل كده يعني عنوان عن الصفة الاخيره بتاع customer
إذا بنارى صفتنيا على الـ constructor customer بتاع لا constructor customer ولا
الـ constructor (name , Id) وانا خارج الـ prototype يعني بيتعاله حاجة أنا بغيره مستحسن
عذان أحل او **Error** ده فروع فورا في الـ constructor customer بـ **no parameterless constructor**

Lecture 6

لما ذكرنا في الفصل السابق هنا خذ وظيفة هنا لنفسها ثم بدل الأوصاف التي من f function بـ f^{-1} عارض صورة .

• ملاحظة ٢: نجد نكتة الـ `display` أخرى بالـ `!important` مطبوع على `name, Id` عارض وذكر `client.display()` بـ `calligraphy` عن طريق الـ `function` `client`

فنت في الـ main لا أكريت order الدينا همتحنونى الفل ولكن مش ده اللي أنا عاوزه
أنا عاوزة الدينا تمسح مخبوط من الـ memory Check بعمل memory صافيه
بعمل destructor بكتب فيه أى حاجة مثلاً وأعمل build هلامتن صافيه
برد جملة مثلاً "Customer is dying" همتحنونه مطبوعه ثلاث مرات.

لدي 9 عملات في المرة الأولى customer will create

كل ده مساحة من خارج memory ملحوظة لبيانات سلسلة سخ من بعض
فـاـلا Customer Create call Chain مـرئـة

امثلة على اساليب توصيل القيم \leftarrow passing by value \rightarrow passing by reference \leftarrow امثلة على اساليب توصيل القيم

للتزم ناخد بالنا اى عنى لغاب دلوقة Copy constructor 2 Cases أستغل فهم

Account<int> s2 = s1; Account s2 = new Account();
order 11 Client uses class with same name as class Pass 12 -- 11

order II Client Name does not have valid pass for card - P
customer.C=Customer(1, Hanif);
order o1(1, 2, 3);

order of (1, 200, C) Copy Constructor خواص جمل

Lecture 6

إنما يجعل البرابر من الشئ المنشئ للشئ المحبب عنه (n)
 heap or stack ويدرك ما يدركه Client II

Customer *C = new Customer (1, Hany);
~~Heap~~ (*) Client I contains Client II which contains order

وذلك بالذات Access II display
إذاً صفاتي هي طاعن (Client II) Chain II يبعث فتاوى به من يعلم نسخ

order (int - id, float - price, customer * - client).

هذا في ماحظكم انه لا يموت ال Client باربع عياد حطين في الا delete main

delete also heap object cause it's in the heap
 معلومة مدهنه \rightarrow لو عملت delete على heap object \rightarrow داخل ال Main

distribution

هذا الحال لو جزى عنهم او heap وجزى من الا stack في حقل

Case 1: passing by reference

Case 2: passing by value

Association - 1

Aggregation - 2

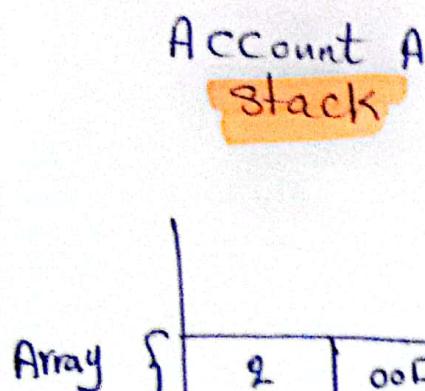
Composition - 3

في حالات متباينة فيها امثل Passing (العنوان ومتى) الحال الأول ومرات من فتح

وهي الحال value or Passing

Passing by reference
aggregation

Passing by value
composition



Heap

type	withdraw	0.0c
amount	50	
type	deposit	0.0f
amount	200	

Account A

int number;

float Balance;

Transiaction Trans [3]

intTop; ← هو شر

كما في المثال لـ Account class
 أهوا معاه الـ Transaction لـ أنه مرتبطة بـ موجودة
 فلدين اثنين على الـ array كل مطابق القيمة
 وبـ destructor وـ موته في صيغة الـ
 باستخدام الـ for loop يعرف مكان كل خانة
 عن طريق المطابقة Top

lecture 7

Inheritance

①

• نوضح بعض النقاط حول الـ Copy constructor

• ايه حالات يستعمل فيه؟

1- 2 objects من Assign العمل

الـ Assign الماء لـ Customer $C = C_1$; \leftarrow object \rightarrow object اسوان

C_1 بمحضه من C ومحضه من C Copy constructor

2- \leftarrow function passing by value \rightarrow function passing (object) العمل

يكتب في \rightarrow Constructor

order(int-number, float-price, Customer-C): Cut (-c)

(Copyconstructor) \leftarrow الماء لـ view \uparrow
(passing by value)

3- \leftarrow function no object \rightarrow return by value العمل

يسعى Copycons \rightarrow return B;

ما هو؟ \leftarrow العمل من نوع object هو هيكل يوصله order no object

order(int-n, float-p, Customer-C) \leftarrow customer

كما في كتب Constructor (من بياض حاج

الـ constructor) \leftarrow مدرس موجود (عندي تالت حلول)

Customer () \leftarrow default override بلدي

4- \leftarrow ما هو؟ \leftarrow customer بكتب Constructor بناع

Customer \leftarrow constructor

فجعل كده! \leftarrow customer constructor \leftarrow object

order(float-price, Customer-C): Cut (1, Harry)

لنفس ازاي \leftarrow Customer constructor

5- الـ customer عرض عليه ازاي \leftarrow عمل لستة بدلالة سمعة ازاي

order(float-price, Customer-C): Cut (-c)

. order(float-price, Customer-C): Cut (-c) \leftarrow يطلع من الـ copyconstructor

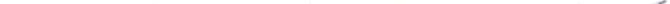
Writing 2 Constructor also new constructor is

Constructor II can be overriden as default parameterless constructor

(3 cases) when is it used - الحالات التي تستخدم فيها

Theirselves

مراجع کتاب معرفت عللات Inheritance also lectures is also available and Full details !!

new class inherit original data (Inheritance) 

base of parent or class II and base classes II can be
driven by child based class

(function , data) (الجهاز المركب)

Child is a parent

وآخر معناه إليه يرجع الـ features لـ مفهـس parent اـنـا بـعـدـا

نحو الكود Inheritanceal account هو في النهاية Personal Account

- Composition or Inheritance

الـ has a علاقة بـ Composition Class A هي جزء من Class B في المفهوم المطابق

جامعة بحثية في الـ Class Inheritance: يكتسب الأفراد صفات والـ Classparent.

تانياً ابداً نعمل على child أول جيله من child فعل فيها الـ Inheritance والـ functions التي هي المقتنيات.

Inheritance || $\text{class child : parent}$

التوريث بضم نباتات أشكال

١- أدوات ليس بيور تتعديل أو إضافة زيارة (أخت) لا child مجهز بـ لعل مستقبل (كوفي تتعديل)

١٥٩٢ - child مولود بورت حنا عليه الناسك كنت أنا عانا نسمونا الألة ١٩٣٩ - ١١٢٠

لصفحة **Attributell** الخاص بي

٣- أعدك على الـ `parent` من الـ `parent` علناه في الآخر لـ `display` function

مکانیکیں شغالہ کو اس عدالت و استخراج میں ایسا وقار حاصل کرے گا جو اس کا ملکیت ہے۔

أخطىء سُفل الـ display .

في حالة التوريث يسدون اضافة أو تغير

مقدمة في C++ | دورة برمجة C++ | دورة برمجة C++

• فتح بـ `new` على Constructor أو Compiler Constructor معاشرة (Static Constructor) موجودة في Class `SavePanel` داعياً `init()` `Errors` `awt`

Account (Account U/{}) is used for maintaining financial transaction.

Account(): Account() {} ← Compiler یک دلیل ایجاد شد ←

Account II : Account() + datatype II من نفس الـ 2 Constructor - بناء المركبة من points chain

PAccount U : Account U { } \leftarrow **define datatype w/ 2 constructors**

• في حالة إنشاء علّي Constructor بابدّه خالص او child Constructor

كتبه. بايد
PAccount(1) : Account(1,200)
الى يأخذ قيم
constructor(2) \rightarrow dublance

PAccount(int _n, float _b) : Account (_n, _b)
سے private || Access → child || child's account ↪
data

ما هو مفهوم child constructor وارقام child يفتح أعلاه Parent constructor

٤ طبعاً نحن عازم قيمه (garbage) بذات الحال اونها هي قيمه افتراضية

PAccount(): PAccount(0,0,0)

وہ میں سچے اکتوبر
اکتوبر

القمامة \rightarrow garbage

١- ساخت قسمتی از main

٩- واحد من main بياخد قيم من الـ

default Inheritance mode

Inheritance in Structs, Classes

- لابيل الـ Struct (Struct)** : هي تابع مفتوحة وهي مغلقة أو مدرفلات
 - إيجاد الـ Parent class (Private)** : لزج على كل التوابع التي في الـ Class

Class Account : public Account from Error Making Public

بعد ما ورثت بس وعرفت تفاصيل التواصي كلها دلوقتي نورث ونمرد_attribute الجديدة

اولیٰ: صمیف اور `private`

ثانية يمكننا أن نلاحظ أن Constructor قادر على إدخال أطليه بداخل him PAccount(int - nid): Account(0,0) حيث كل الـ Attributej

وَهُكُمْ أَعْمَلُ وَأَدْبَرُ مِنْ أَحَدٍ حَاجَهُ لِنَفْسِهِ الْكَيْفِيَّةُ

ويمكن العمل واحد بأخذ كل المعلومات من المرا

`PAccount(int-nid, int-n, float-b): Account (-n, -b)`

أقترب أعمل أي سُكّل براحتي المفهوم أكون فاهم كل واحد استخدموها عليه.

و زیرا ما عملتی نهاده همچنان `Attribute` هرچوی هم `display` باشد `PAccount` را باز خواهد داشت.

فيها البرز الجديـد بالشكلـه

و لكنها اثنان من الاختيارات

ويمكننا عد دائرة π بـ $display$ بآيات الآيات

٢- عارف أطبع النلات فهم لهاردي على (Attribute) (Attributed to)

- ملعوا في سطع واحد ليه حمل آردو ۹

الناء الطيارة كل مجموعة ارتبطت بمحضر  Access Child آدأي أطياف

• **private** (بینکتی) + Access Modifier
• **protected** (پوشیده) + Access Modifier
• **public** (عام) + Access Modifier

ـ أقر، Accessi بطرق تسمـ data

ـ بـ طـرـيقـه عـرـجـيـهـاـسـرـهـعـرـطـرـيفـهـإـنـأـخـلـChamـبـسـالـConstruct or (parent, child)

ـ بـ طـرـيقـهـمـيـاسـرـجـلـسـتـخـدـمـالـAccessـ الجـديـدـModifer

ـ وـ جـاـإـنـقـرـتـاـهـمـعـرـكـتـهـاـلـdisplayـبـلـاتـ

ـ child
ـ void Display ()

ـ كـوـدـهـيـمـكـتـسـأـقـرـالـتـيـهـمـاـلـأـوـلـ
ـ لـدـمـ الـNIDـ وـ الـBalan~ceـ وـ الـNumberـ كـانـواـ
ـ بـالـنـسـابـاـلـيـ اـحـاـ حـالـيـاـ
ـ pri~vateـ لاـ

ـ باـكـرـقـمـنـهـ هـقـدـرـيـ (ـنـيـ أـلـإـلـإـلـ)
ـ مـيـنـفـعـنـ الـبـنـهـمـ فـيـ جـزـءـ الـPr~ivateـ الـchildـ فـيـمـ

ـ لـأـنـ لـزـمـ أـفـهـ الـP~arentـ مـاـهـوـ الـأـصـلـ لـوـعـلـيـ
ـ فـيـ هـبـنـظـ الـدـنـيـاـ بـعـدـهـ وـلـيـقـعـ الـقـيـ الـC~hainـ

ـ كـانـ لـلـزـمـ أـحـفـظـهـ الـC~hainـ مـعـنـاتـ أـنـهـ

ـ الـكـوـدـ مـكـوـبـ مـكـانـ وـاـحـدـ خـفـطـ وـ التـعـيلـ يـكـوـنـ سـطـلـ

ـ اـنـزـالـ الـInheritanceـ وـعـلـىـ الـPoly~morphismـ الـعـيـنـ

Inheritance وملنالا

True polymorphism

احنا شفنا الـ polymorphism بالذرة من شكل سواء عن طريق **overriding** أو **overloading** ولكن الا الامثلة دى مكافئتين بتوصلنا للـ **maximum Polymorphism** بناء على المنهج **maximum**.

والـ **maximum** بقى هو اى اعمل **function** هبيغتبطها **Array** صدفه رايمد فله وهي نفس نوع الترميز نوع وبالنال كل نوع هتسنجل عليه بسكل مختلف إذا لا أنا ولا الـ **Compiler** عارف فيه النوع اللي هيبيجت ولا عارفيم هن هتسنجل بنادأ على اييه ده صرفة الـ **polymorphism**.

سوية Steps توصلنا للـ polymorphism

❶ ن أقبل الـ **Account** والـ **Personal Account** المنظر يقول في مينفعش أعمل **Account** فقط أو مينفعش أعمل **object** مدنوع **Account** فقط ليه لا بد مننظم **object** أنا كل مرجم يا هعمل حساب شخص يا حساب شركات ولكن كـ **Code** أقدر طبعاً أعمل **object** ولكن من الواقع مدهش هتروج يعنى حساباته نافض معلومات.

❷ لو لقيت كود زى ده هيعدى ومن صيغته 5 Errors

Account A₁ = personal account (2,200, 2222)

لـ **يه** هيعدى بـ **ي** لـ **ما بجريت** **object** من نوع **Account** هيكو هكون منه جزءاً من الـ **memory** والجزء تمام دول موجود بـ **الان هما (2,200)** هو من عاوز التزهد كده اللي حصل ده اسم **slice** (يعنى يأخذ شريحة من الـ **data** بما يتاسب مع ما ياقظ).

لو جربنا انظر **slice** مع الـ **variables** من الـ **object** (هكله نفع وهملاً)

مثال int x = ...

الـ **كود** ده هيعدى في حالة إن **حبيحة** الـ **(X)** هكله تتشال في

slice فقط وهو ده سمعنا الـ **إنا لا الـ **Byte** 1** من هينفع يصل هستقولي ما بعيت الرقم الزبارة بترى عارف زعماً ما حصل من الـ **Account** هقولك إن الـ **Account** ده **Object** هو أساساً جواه مكان لـ **رسيل** معلوماته إنا لا واردها أملكه لـ **رسيل** معلومات واحدة فقط لازم المكان يلتف المعلومة رو.

الـ **حصل** في مثال الـ **Account** إنى خليت الـ **personalacc** الـ **Account** همنته

حصل تحويل ده **datatype** إلى **datatype** آخر ده **Account** على إنا عدل نوع **data** كـ **ان** بقوله شوف الـ **Account** فقط

lecture 8

٥

٣) المكتن غير صحيح بحثه لوليت

Personal account PA

=A CCOUNT(1,988);

المؤود > ٠ هفڑب Error لیه؟

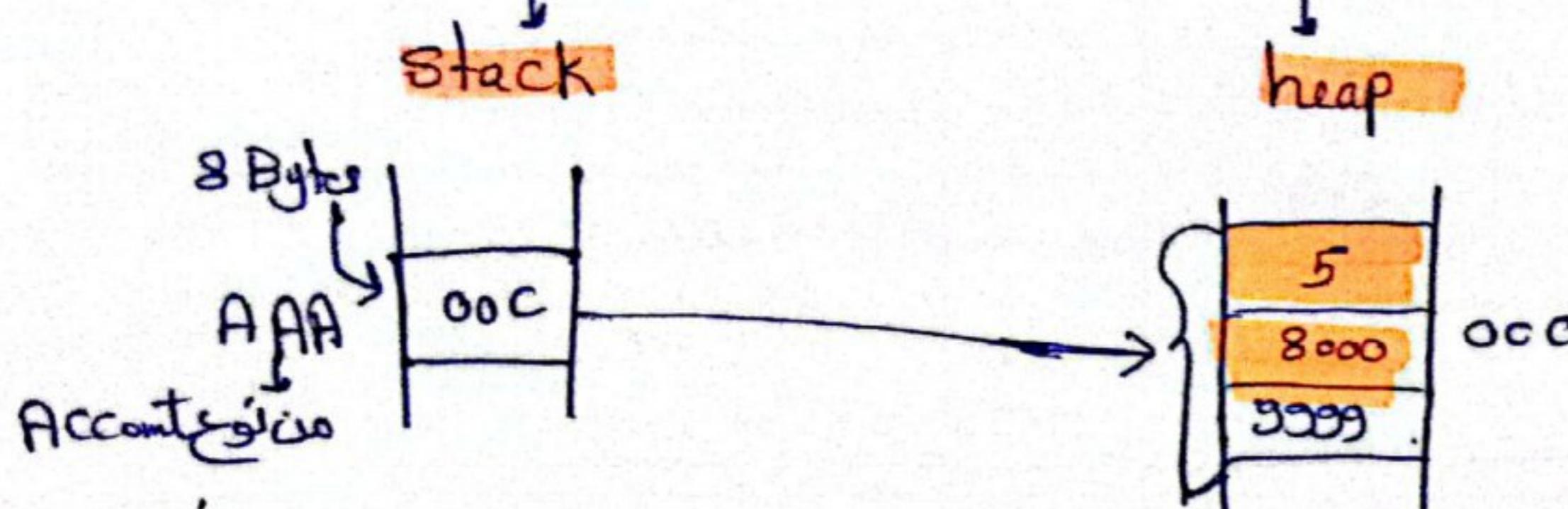
لِسْمَ الْأَرْبَعَةِ personal account هو هستي ۳ معلومات والـ مكتوبته ليس إداً هقد (بـ) الـ object فيه معلومة ناقصة.

٢) لما بعمل slice فـ الا الـ stack بيتبيل بس القسم اللي هو محتاجها والباقي لازم ما يستقى

ـ إنما لو حطينا القمـ اهـلـاـزـ منـ الـ هـيـلـاـزـ لـ يـهـ؟

لابد أن heap مساعي من حيث المساحة \Rightarrow هذه المساحة تسهل تسليل المجزء الثالث

Account * AAA = new personal account (5,8000,9999)

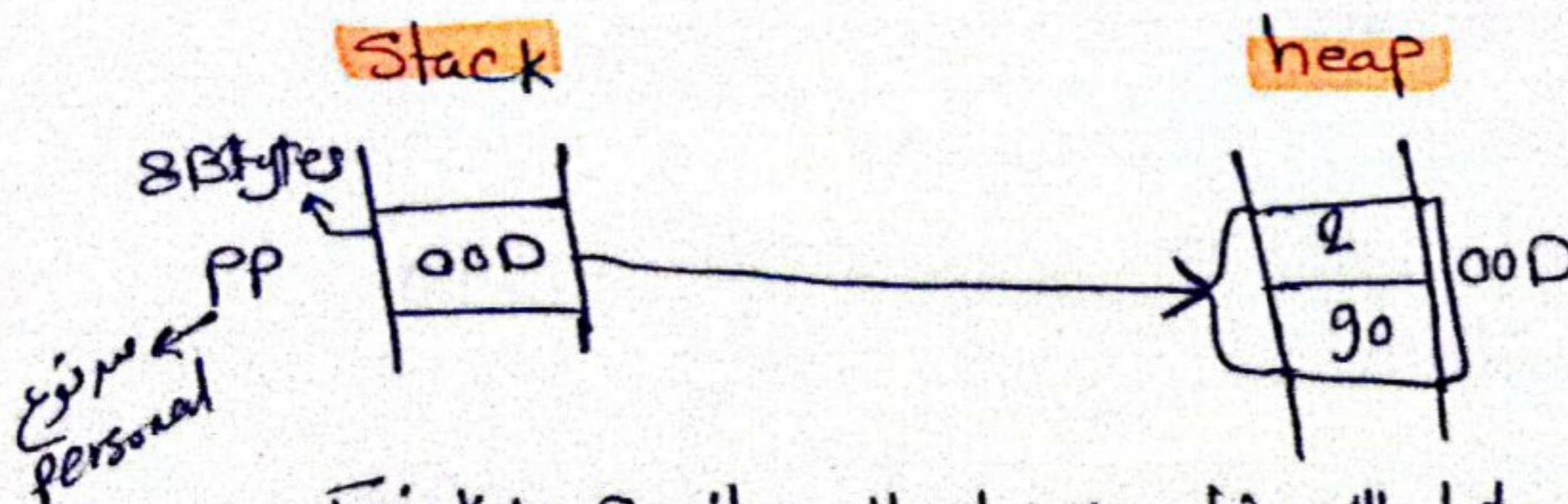


الـ `AAA` هو `pointer` يُسَمِّي `Account` data المكونة من ٣ أجزاء ولكن سلطته على أول قيمته فقط لأنها من نوع `Account`

إذا أقيمة الثالثة هي بالفعل موجودة في الـ heap لكن هل هناك قادر **Account** أو مثلاً
أو أشخاص فرما لا . وبالنهاية يظهر لي أول قيمة فقط . 15,8000)

٥ الديكس غير صحيح لوكست

Personal Account * PP = New Account (2,90).



المفترض هنا أن المجلة دى
ترجعى بلاغى كده كده
pointer بـ بساور
والألى pointer مكانت
عن الـ memory معروف

فـ الـ ^{Memory} مـ رـ وـ فـ الـ Compiler هـ نـ حـ مـ نـ لـ كـ دـ هـ ؟
أـ زـ يـ كـ اـ نـ تـ جـ حـ مـ الـ data الـ لـ بـ سـ تـ لـ وـ عـ لـ يـ حـ اـ . لـ يـ الـ Compiler هـ نـ حـ مـ نـ لـ كـ دـ هـ ؟
يـ لـ يـ مـ سـ تـ قـ بـ لـ آ مـ كـ لـ مـ أـ مـ لـ يـ هـ مـ نـ function تـ حـ لـ شـ غـ لـ عـ لـ الـ فـ تـ يـ حـ ةـ (i) وـ فـ حـ مـ نـ الـ بـ اـ يـ
بـ يـ مـ نـ حـ دـ كـ تـ حـ لـ شـ غـ لـ عـ لـ الـ فـ تـ يـ حـ ةـ (i) وـ فـ حـ مـ نـ الـ بـ اـ يـ

lecture 8

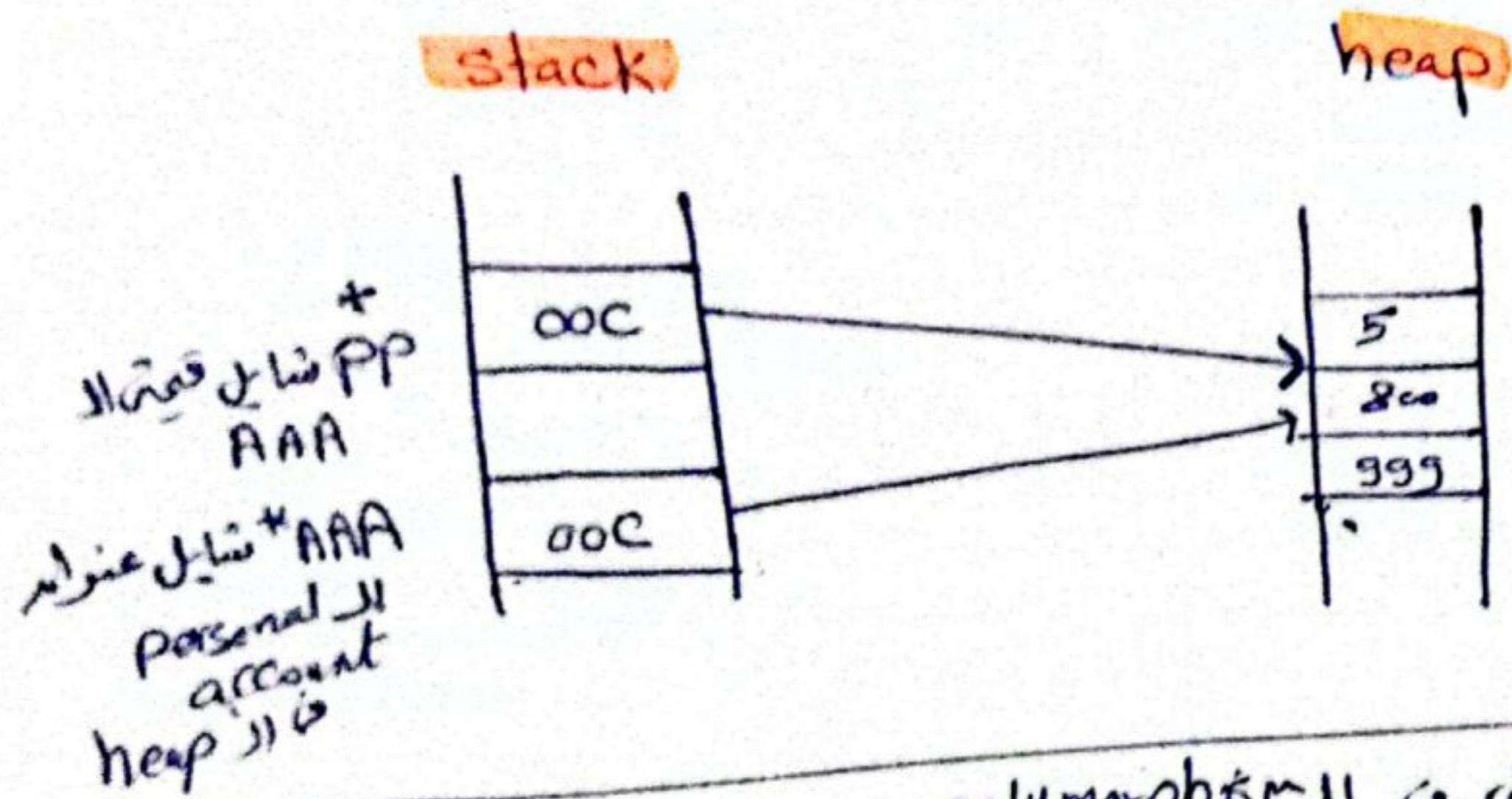
`new ooc ← personal Account * PP = new Account (2,90);`

`new ooc ← Account * AAA = new personalaccount (5, 800, gggg);`

فإذاً يمكن أخذ الكود الأول الذي هو كالتالي

`personalaccount * PP = (personal Account *) AAA;` أعمل كده

كما يسأول لك Compiler أي Account هو فعلاً من نوع personal account؟ ثم في الورقة هناك مثلاً فيه heap ذي نوع personal account.



الخطوات التي تاتي هنا هي steps من polymorphism.

نرجع للطريق الذي نسبت عاوزه أوصله إلى أعلى function بطبع function ابعتلك معلومات أى حد سواء Child or parent تستعمل.

عسايده كده أول خطوه هي توحيد أسماء الـ functions بطبع للأب والابناء.

Child ← Display → Parent

بس أنا عارف إن الـ Compiler كل حرج هيرجع لبعض الآباء بين لذلك كان لازم نظر

إلى display بتعت الآباء تستعمل ← عذر طبعكم كبيه قبل قلم الـ virtual void Display ()

وريكيه أى كان الـ object اللي بيستاده على display من نوع اللي بيستاده child أو parent.

lecture 8

四〇

array II, Job

```
int x[7] = {1, 2, 3}
```

block of elements

array II jei is.
badArray II: will
the same datatype

Lambeth

اما من حصلت عليه الامانة فهو ملوك جواه اكتر من نوع datatype وهو ده المطلوب حال Case

لطفاً داده و داده Datatype ایک ریکارڈ کے لئے مخصوص ہے۔ اسکے مکانیکی طور پر اس کو دیکھنے کا سب سے بہترین طریقہ ہے۔

اما مفهوم الارض فهو معلم يكتبه جواه اكتر من نوع datatype وهو ده المطلب فال

(parent) Account ونوع dataTypes array جزء

```
Account* tarr[ ] = new TPersonal(1.5,30),
```

الكور ده معناه انه ظاهرتنا الـ *display* بتاعة *array* *Account* دلخه ده أصلان بالـ *Steps* { واهنا عرقاليه الكور هيعدى من الـ *array*

Corporate account (2.6.1b)

2

3

أنا خلاص حزت الـ function الـ `DisplayAll()`  بـ `array` انتارى على الـ `DisplayAll()`  هنكل السفل ده

array دی دستیبه مخلوحته ۱- ای ای جوا ای ار elements ۲- عدد الی جوا ای ار

⇒ Displayall(arr, 3); This

طريقه تعبير الـ function

{ for (int i=0; i<size; i++) } array العمر فيه بيتدى من الصفر
فقوله طول ما ازا (n) اقل من الا Size اللى اصو

{ arr[i] → Display(); } } *out.println(element);*

(Generalization) \leftarrow اینجا \leftarrow True polymorphism همودهول

lecture 8

التجدد من التفاصيل Abstraction

Abstracted function: function without body.

parent class مبتدة عن العالم والتفاصيل تكون موجودة في function وتسماها -١

Pure Virtual Function

virtual void withdraw = 0; طريقة كتابتها

abstracted function

لديه عنى ٢

عستان يمكنه عنى function بطرق مختلفة child ولكن ملاش وظيفة حقيقة عند Parent عشان كده بتكتبي فقط بعد تفاصيل او body

طبعاً ما ممكن أستخدا من الأشياء فقط إيه لازمة كتابتها في الأنباء

إحنا عرفنا اتنا لما نعمل الـ Standalone function الأشياء اللي يقدر يشكل شكل الابناء هتش الكلمات

وإنه إحنا اتناء كتابتنا للـ Parameters هنقول ما هو كل مرة هيسوف بتعتبر الـ Parent اللي من بتعل حاجة همولك لا بد في كيمنت قبيلها virtual (لذلك يجب كتابة جميع الـ functions بنفس الاسم في الـ parent ولا child)

محظوظ وجود data type في abstract function ، abstract data type

concrete data type

complete Data type .
أفتر من خلايا المروجت object عارى

abstracted data type

غير كاملة مقدرش من خلايا Data type ←
الكريت object لا يحتوا على الـ abstracted function .

الـ Compiler هيمنعن من الداير إن الـ object لأن هو هيمنعن إن هنكت استخدمو الـ abstracted function دى ولا ولا ولو استخدموها هيلا فيها من بتعل حاجة أصلها في من البيانية من هيكلتك تعلق object منه الأساس .

Lecture 8

Dynamic polymorphism

لهم أنواع الـ Polymorphism ⑦

Static polymorphism

Static polymorphism

It occurs at Compile time
Line اثناء تراجمة Compiler الـ (يجعل الـ Compiler يختار)
function الـ (يحروف الـ Function) Line by Line by

(لـ تستدل ازاي او ايـالـهـيـيلـاـ)

early binding . (اسم ارضـاـ)

(Function) - Method overloading
operator overloading . (اـشتـارـاـ)

operator overloading (لا تفضل علىـ)
operator (وبـ الـ وـ بـ الـ)
operator (وبـ الـ وـ بـ الـ)

Readability (تـخلـ منـ الـ)

Dynamic polymorphism

It occurs at Runtime

Compiler الـ (يختار) وـ دعـوـيـقـرـاـ

عنـ قـيـقـةـ عـارـفـ (يـهـ الـ فـيـعـيـلـاـ)
ولـ طـبـيـعـةـ شـعـلـاـ .

late binding . (اـرضـاـ)

(Function) Method overriding .
operator overloading . (اـشتـارـاـ)