

Lecture 4

Default Parameter less: a constructor (a gift from compiler)

اللي هو لو انا ماحطتش constructor يعني هو هيحطه .. بس قيمته هتكون بـgarbage

لو حطيت انا الـconstructor خلاص بقي هنتلغي الهدية

Class is **private** by default

Struct is **public** by default

Instance data

التغيير فيها مش بيأثر ع الباقي

Static data

- عندي مثلا معلومة هغير فيها و هتسمع مع كل النسخ (objects) اللي عملتها
 - variable مؤقت (يحط لي القيمة)
 - المعلومة اللي تكون static بتعامل معاها من خلال data type مش خلال النسخة (object)
- فيعني اللي ينده عليها الـ data type مش النسخة..و بالتالي ده معناه ان ماينفعش جواها >-this .
بالمنطق كده مالهاش لازمة جوه الـ static function

Destructor is a special function and has the same name of the data type.

It does not have access modifiers (it is public by default).

(telda operator) ~ دي بتبقي قبل اسم الـ data type

```
~trainee () {
```

```
}
```

يبقي دي فانكشن ليها نفس اسم الـ data type و هتموت بعد اخر حاجة هتعملها ف نهاية الـ object

- لو في الـ stack هتموت بنهاية الـ scope {}
 - لو ف الـ heap هتموت لما انا اطلب كده بـ delete مثلا.
- It cannot be overloaded

Why Destructor is existed?

It is **cleaning up unmanaged resources** (fields) in the operating system.

هي حققت المطلوب كده كده .. علاقتها بالـ OOP بقيت بكتب و بعدل ف مكان واحد بس

→→

Reusable & Maintability

Friend Function

بتكون معرفة برة الـ data type بس تقدر تـ access الـ private بشرط ان الـ data type هي اللي تكون قابلية كده (سمحت لها بكده يعني)

يعتبر عيب ف الـ C++ لانها بوظت مفهوم الـ Encapsulation

Friend void DisplayOnlyNumber ();

Lab 4

- 1- Create Data Type That Encapsulates Account Data Type With Only One Constructor That Takes the Balance only and Increment the Number With One.
- 2- Create Display Function That Display Account Data (Number, Balance)
- 3- Test Your Data Type With Two Variable Of Bank Account.
- 4- Instance Data vs Static data With an Example Of Your Own.
- 5- Exmapline What is meant by friend function.
- 6- Bonus - What is inline function?

Answer

```

1  #include <iostream>
2  using namespace std;
3  class account{
4  private:
5      static int Count;
6      static float interestRate;
7      int number;
8      float balance;
9      int id;
10 public:
11     static void setcount (int _Count){
12         Count = _Count;
13     }
14     static void setinterestRate(float _interestRate){
15         interestRate = _interestRate;
16     }
17     account(float _balance){
18         this->number= ++ this->Count; //increment (++1)
19         this->balance= _balance ;
20     }
21     friend void displayOnlyNumber(account A);
22     void display(){
23         cout <<"number"<<number << endl
24             <<"balance"<<balance<<endl
25             <<"interestRate"<<interestRate;
26     }
27 };
28 void displayOnlyNumber(account A){
29     cout <<"number"<<A.number;
30 }
31 int account::Count = 0;
32 float account::interestRate = 100;
33
34 int main()

```

```

34     int main()
35     {
36         cout << "enter count:"<<endl;
37         int mycount;
38         cin>> mycount;
39         account::setcount(mycount);
40         account a= account(2000);
41         account b= account(5000);
42         a.display();
43         b.display();
44         displayOnlyNumber(a);
45         return 0;
46     }
47

```