

2) object :

لما نحتاج نستخدم الـ class اللى عملناه أو نستخدم أى حاجة جواه نعمل
منه object فإمكان ده وليكن داخل الـ main

```
void main() {
```

* `classname + object name = class name();`

→ إنشاء `User user = User();`

* عشان استخدم اللى جوا الـ class

يتم استدعاء الـ constructor
الـ default

→ `user.function();`

→ `user.variable;`

Constructor :

ده عبارة عن method بتنفذ بكل تلقائي اول ما نعمل انشاء
لك object زي كده `User user = User();`
وبما ان الشكل ده ما فيش parameter بيعوت مع الاستدعاء فده
هيقم تنفيذ الـ constructor الـ default وبعينه الـ constructor ده
ان حتى لو احنا ما عملنا هوته هو بيعمل معول جاهز.

لكن لو `User user = User("mahmoud");`

الشكل ده هيقم الـ constructor اللى الـ parameter جايه string

الشكل ده
مجان تنفيذي

```
User(string name) {  
    username = name;  
}
```

→ `User(this.username);`

الحالتي دي لو انا عايز استقبل
variables بس ومنه هاعمل اي عملية

لو `User([this.username]);`

معناها ان الـ parameter هنا اختيار

2) object :

لما نحتاج نستخدم الـ class اللي عملناه أو نستخدم أي حاجة جواه بنعمل منه object فإمكان ده وليكن داخل الـ main

```
void main() {
```

* $classname + object\ name = class\ name();$

→ إنشاء `User user = User();`

* عشان استخدم اللي جوا الـ class

نستخدم الـ default constructor

→ `user.function();`

→ `user.variable` ;

Constructor :

ده عبارة عن method بتنفذ بشكل تلقائي اول ما بنعمل إنشاء

لـ object زي كده `User user = User();`

وبما ان الشكل ده سايفيش parameter مبعوت مع الاستدعاء فكله

هتتم تنفيذ الـ constructor الـ default وحينئذ الـ constructor ده

ان حتى لو احنا ما عملنا هوته هو بيعت معول جاهز.

لكن لو `User user = User("mahmoud");`

بالشكل ده هتتم تنفيذ الـ constructor الـ parameter جاي string

```
User(string name) {  
    username = name;  
}
```

الشكل ده
مجان تستدعي

→ `User(this.username);`

الحالتي لو اتا عايز استعمل

variables بين ومسته عامل اي عليه

`User([this.username]);` لو

معناها ان الـ parameters هنا اختيار

بنفس الاسم ۵۵

② User. { } () ای اسم
تانی

```
User.secondCnst( ) { }
```

→ في العادي `User.user = User();`

الجدید بقى → `User user = User.secondConst();`

```
↳ User user = User.NamedConst();
```

③ encapsulation:

* اي حاجة داخل ال class عشان اعرضها باستخدام لها
method اسمها getter وعشان اعدل عليها باستخدام لها method اسمها setter
← ال (-) قبل اسم المتغير بـ private بالنسبة للفايل اللى هوفيه ديكده مانيفست
استخدم (انذهلي) باسم على طول User.Variable ولازم استخدم ال getter وال setter

```
string? _Password;
```

استرجاع كلمة المرور: `User.getPassword()`

```
⇒ getPassword() { ←
    return _password; }
```

```
⇒ setPassword (string word) {
```

- Password = word; }

استدعاء `User.getPropertyname`؛

getter:

```
return type get propertyName { return _variable }
```

" " " " => Variable ;

setter:

```
set propertyName (value) {
```

```
set Password (string value) {
```

```
    _Password = value; }
```

الاستدعاء

```
user.Password = "La vie 666";
```

نمبر الصلة بعلامه (=)

(4) Inheritance:

لومني متغيرات او methods مشتركة بين اكثر من class مخططهم من class Parent ونجلي المترين معاه children يورثوا منها

```
class Parent {
```

```
    string? name;
```

```
    string? email;
```

```
    int? phone;
```

```
    getEmail() {
```

```
        return email; }
```

```
}
```

```
class child extends Parent { }
```

امسح صتيو كل ما بداخل او Parent

```
child ch = child();
```

لما نخل object

هتقدر نستخدم اي حاجه جوا
Parent class

لو عملنا class ثاني بي extend او child فصيقي من خلاها بي extend
او Parent الاكبر وبتستخدم كل اللي فيه برضه.

Hierarchical inheritance: Type of inheritance where multiple child classes (sub classes) inherit features from a single Parent class (superclass)

Constructor:

لو مني child class بيورن من Parent عند عمل object من انا child
بيتم استدعاء انا constructor بتاع انا Parent

طب لو انا constructor ده ليا Parameter :

← لو انا من انا Parent نستخدم this.

← لو انا من انا child وعالين نستخدم حاجته من انا Parent نستخدم ^{super}

```
class Parent {
```

```
    string ? name;
```

```
    Parent (this.name) {
```

```
        Print ("name : $name"); }
    }
```

```
class child extends Parent {
```

```
    child (super.name);
```

(اد)

```
    child ({super.name});
```

المتغير موجود من انا
superclass

بني انا مني هشتدعيها
في انا object ك ("name")

• ودلوقتي لازم نمرر انا Parameter عند عمل انا object من انا child

```
Child ch = Child ("mahmoud");
```

Child

Parent

هيعود نيق انا constructor بتاع انا ~~Parent~~ وبيده نيق بتاع انا ~~Child~~

note

* this نستخدمها مع المتغيرات الموجودة داخل نفس انا class .

* super نستخدمها مع المتغيرات اللى بنورثها من انا Parent ومش موجودة

معانا فالكلاس اللى انا مني (كتابته يعني)

لوعائيرة استخدم method موجودة في اد Parent في اد child
يمكن استدعائها على طول عادي (لكن الأفضل احدا قبلها super)

getName() → ممكن

super.getName() → الأفضل

عشان نتعرفنا اسمها بجاية من
اد Parent .