

OOP object oriented Programming

- easy to understand
- Increase reusability and decrease complexity
- Increase productivity
- Code easier to maintain, modify and debug
- reduce repetition of code

Features

- Class
- object
- Encapsulation
- Inheritance
- Polymorphism
- abstraction

## # class

```

    Capital
    ↗
Class Users {
    String? username;
    Password;
    PrintName () {
        Print(username);
    }

```

include variables &amp; functions

→ Properties  
→ attribute

→ method

→ Create object

Void main() {

object consists of #

```

Users Users = Users ();
    ↗       ↘
class name object class name

```

Users.userName = "Mhmd";

Users.PrintName ();

method

## # Default Constructor :-

← تستعمل بكل مكان عند إنشاء object  
 (method name) →

class  
name

Items H={ (string name) {

name = name ;

هنا عند إنشاء أي object اكتب فيه name

Items item = Items ("laptop");

Items (this, name);

← الوجود داخل class ( [ this - name ] )

← اختيار من لازم اختيار

Items (this.Price, [this, name]); ← argument

↓ اختيار  
 ↓ اختيار

Items ( { this.Price, this.name } ) map

Items items = Items (name : "laptop", Price : 200) و (من لازم ترتيب)

هنا اختيار واحد (و لازم اختيار)  
 required

required this.Price

## # Named Constructor :-

لوحنا اسم Constructor من لازم لين

Users.NamedConstructor (this, username);

Users.NamedConstTwo ();

Users users = Users.NamedConstTwo ();

يحدد من يستعمل عند الدخول في

كأن default كد ال Default



## # Encapsulation: (hiding data)

Setter ← لا يمكن الوصول إليها  
 لا يمكن الوصول إليها  
 Setter

class Users {

String name ;

String age ;

String balance; int - balance = 0 ;

Users ( this.name , this.age ) ;

- Variable Name

hide  
Private

File لكن في Private dart من ال Class  
 لا يمكن الوصول إليها

Print ( users.balance ) ; error

Getter

Setter

getBalance() {

return -- balance ;

setBalance(Val) {

- balance = val ;

من الفائل الآخر

users.setBalance("Mhmd") ;

Print ( users.getBalance() ) ;

deposit (int val) { Setter

- balance = - balance + val ;

withdraw (int val) { Setter

- ~~besta~~

if ( - balance &gt; val ) {

- balance = - balance - val ;

} else {

Print ( "لا يمكن سحب أكثر من المبلغ المتاح" ) ;

users.withdraw(100) ;

Print ( users.getBalance() )

من الفائل الآخر

لا يمكن سحب أكثر من المبلغ المتاح

ال user لا يمكن سحب

Balance بشكل مباشر

Best in Getter & Setter :-

get → return type get Property\_name {  
// body  
}

int get balance ⇒ - balance ;  
Print (users - balance);

set → set Property\_name (value) {  
// body  
}

set deposit (int val) ;  
- balance = val + - balance ;  
}

users. deposit = 400 ;