

Inheritance Concept

مقدمة Inheritance مفهوم
وهو "استيراد" لـ "Inheritance" و "Overriding"
وهو المفهوم الذي يتيح لك إنشاء فئة جديدة (Subclass)
تحت فئة أخرى (Superclass).
يمكنك إضافة خصائص و METHODS
جديدة إلى الفئة الجديدة،
ويمكنك استخدام خصائص ومETHODS
القديمة في الفئة الجديدة.

```
class Mobile {  
    String screen;  
    String Camera;  
    void PrintNameMobile (name) {  
        Print(" Mobile " + name);  
    }  
}  
Samsung extends Mobile {  
    void main () {  
        Samsung Q56 = new Samsung();  
        Q56 . screen = "6.4";  
        Q56 . PrintNameMobile ("Samsung");  
    }  
}
```

Get

```
String get username {  
    return " how are you $username "  
}  
("Wael")
```

خطيبVar مثل staticelas كون من
@UserVar جسم ذات يعني من يفتح تغير
Class var . var = ^{فقط} _{الجديد} Var

class User {

static String username = "wael";

String get my name {

return username;

void main () { "Hello" . www . sf.com

User userone = new User();

User . username = "Elham";

Print (userone . my name); ^{outPut} Elham

Print (User . username); ^{outPut} Elham

وهو الذي يمتلك

Mobile mob = new Mobile() - testOne()

Class موجود في جواز

1 Constructor

دی (ریزدستی یعنی هر دو دست)
 void main() {
 const int a = 10; // اول مادرید
 cout << a; // بعده
 }
 mobile mobile's new Mobile ("Elham");

lass Mobile f

mobile (3 screen) of

`Print (screen);` → Constructor

11 Rest Var

1. Mobile mobile snw. Mobile () -- names "them"
-- Phone = "alo--"

- mobile.name = "Elham"); // name bind

- Constructor is like var "jaisa" constructor ka

Mobile (val) of (snowy, snowy) tree set

Screen = Val; { get }
- Set ~~get~~, get

old set changusername(newname) {

this.username = Newname;

Var *class* *is*

oof

Class Mobile of

double screen;
string name;

1

Void main () {

Mobile mobile = new₂ Mobile();

```
mobile.Screen = "5.7";
```

mobile.name = "iPhone 17 Pro Max"

* //method in class

Class Mobile {

double screen

String name.

~~Mobile motarite~~ = new Mobile C1;

Voided Print-Screen (1)

Print(screen)