[Arrays of objects]

The two level referencing

→ it is an array of references
so before the creation the default value of the reference array variable is null and can't be accessed without creation.

→ if accessed without creation → (null exception) runtime error ✓

→ if the array is called (student[])
each object will have the name student[i] and every thing relating to the object inside it will be accessed by this name. ✓

[Immutable objects]

refering to the object whose contents cannot be changed once that object is created.

immutable object ⟷ immutable class

✓ nothing except private datafields
✓ no mutators
✓ no accessors for references

→ for immutable classes, the string is allowed to have an accessor (getter) regardless it is a reference because basically it is not mutable ✓

مش لازم يكون في access ع أي reference بأي شكل
يعني مثلاً انسخ الـ address الي في الـ reference
تاني، لانه هادا الشي راح يعطيني access عليه ✓

[This keyword] ← بيرجع الـ reference تبع الـ object ✓

← بقدر استخدمه داخل الـ class مثل الاسم تبع الـ object نفسه ✓

① it is used to access hidden variables

But

- instance variables are accessed by it [Hidden ones]
- static variables are accessed by class name and no need to use this [when hidden]

② Used to invoke a constructor

أحياناً بنحتاج إنه ننادي constructor جوا واحد ثاني، بسبب تشابه وظايفهم مثلاً، فبننادي باستخدام this مع وجود شروط:-

① ال invokation ← this(arg-list) لازم تكون أول سطر في ال constructor قبل أي statement ثانية.

▸ Subject : ......

[Method Abstraction + Stepwise Refinement]

[Abstraction]

ما خفي implementation الـ
use الـ

معروفة كان باسم :

Information hiding or encapsulation ✓

The client code use a method without knowing how it is implemented ✓
→ only needs to know what it takes in the arg-list and what it returns

* you can change the implementation of the method without affecting the client code [because the job of the method is still the same and you do not change the signature]

ubject : .....

[Stepwise Refinement]

also known as divide-and-conquer strategy ✓

the large program is decomposed into subproblems, and the subproblems themselves can be decomposed into smaller more manageable

[Special Study] ⇒ Class Abstraction and Encapsulation

(1) The user side :-

→ user does not need to know how the class is implemented

→ details of implementations are hidden from the user and this is known as [class encapsulation]

Thus, the class is known as [ADT]
Abstract Data Type ✓

Subject :

② creator side

→ the creator describes the function of the class to the user (and how the class can be used)

[class contract]

signature of public methods and fields and constants that are accessible from outside the class [with the description of how they are expected to behave]

Subject : .....

Object-Oriented	Procedural
[1] Focuses on: → coupling data and methods together into objects [Software design using object-oriented paradigm, focuses on objects and operations on them.]	[1] Focuses on: → designing methods
[2] Data and operations on them: → can be gathered in objects → can be reusable codes	[2] Data and operations on them: → are separate, requiring data to be passed to methods. → cannot be reusable until it is (the code) put in a method (static) the reason they are not reusable is that the code is in the main
[3] limitations: (no limitations) → data are coupled by creating objects (individual objects	[3] limitations:- → you cannot associate declared things with some defined objects variables can be declared but not coupled under meaningful object

Scanned with CamScanner