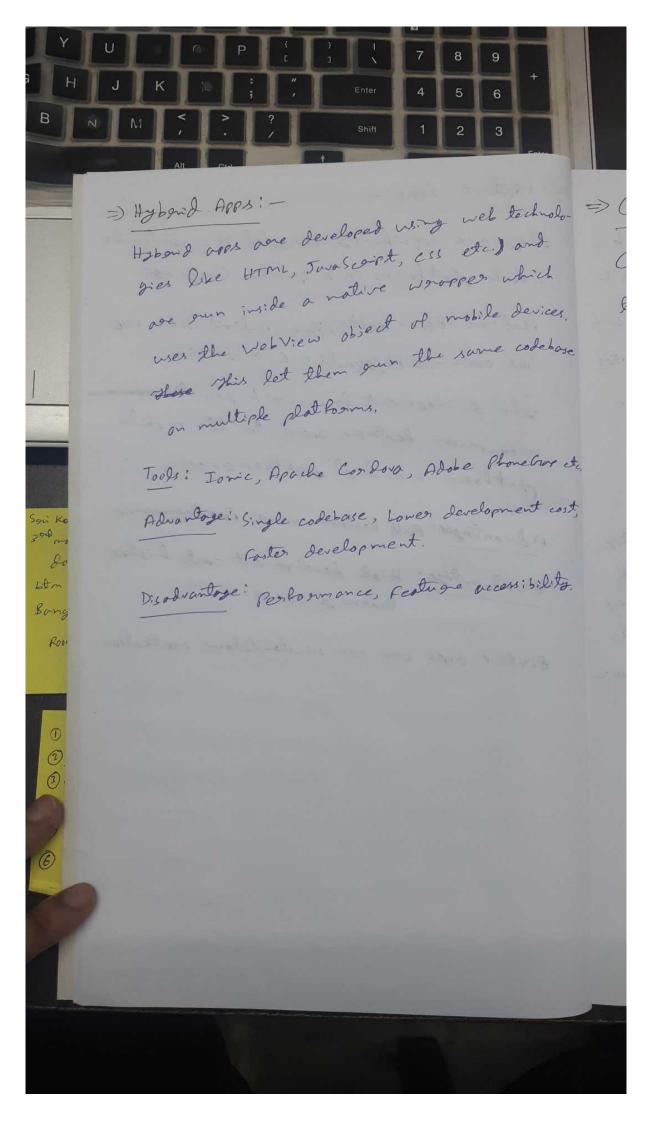
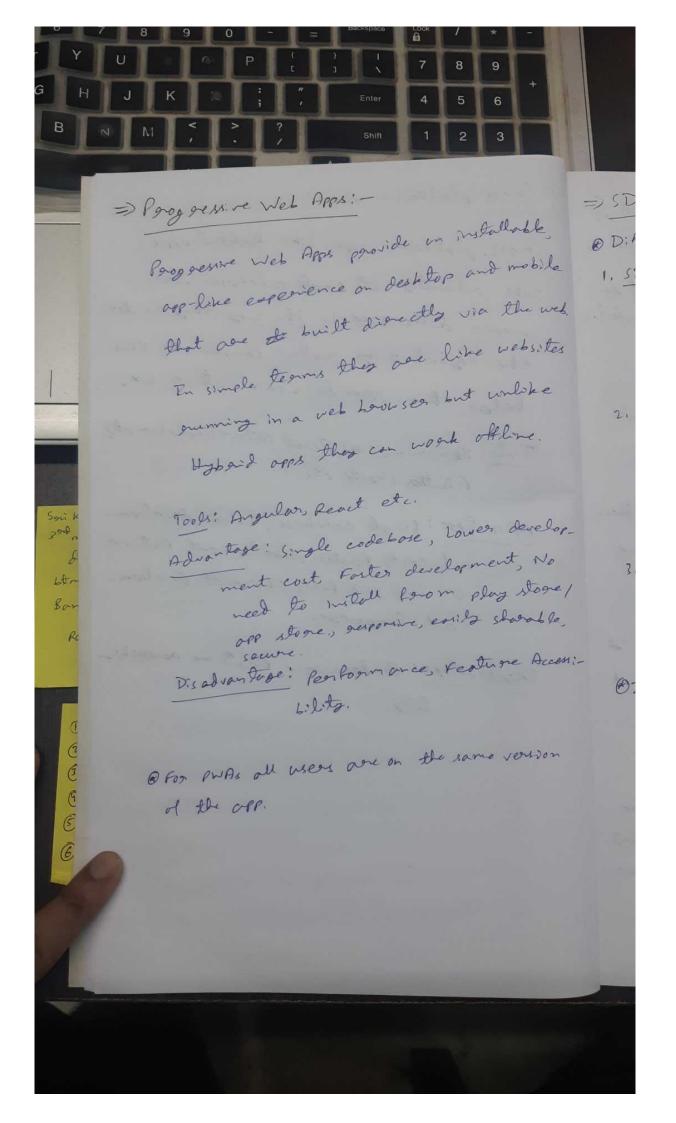


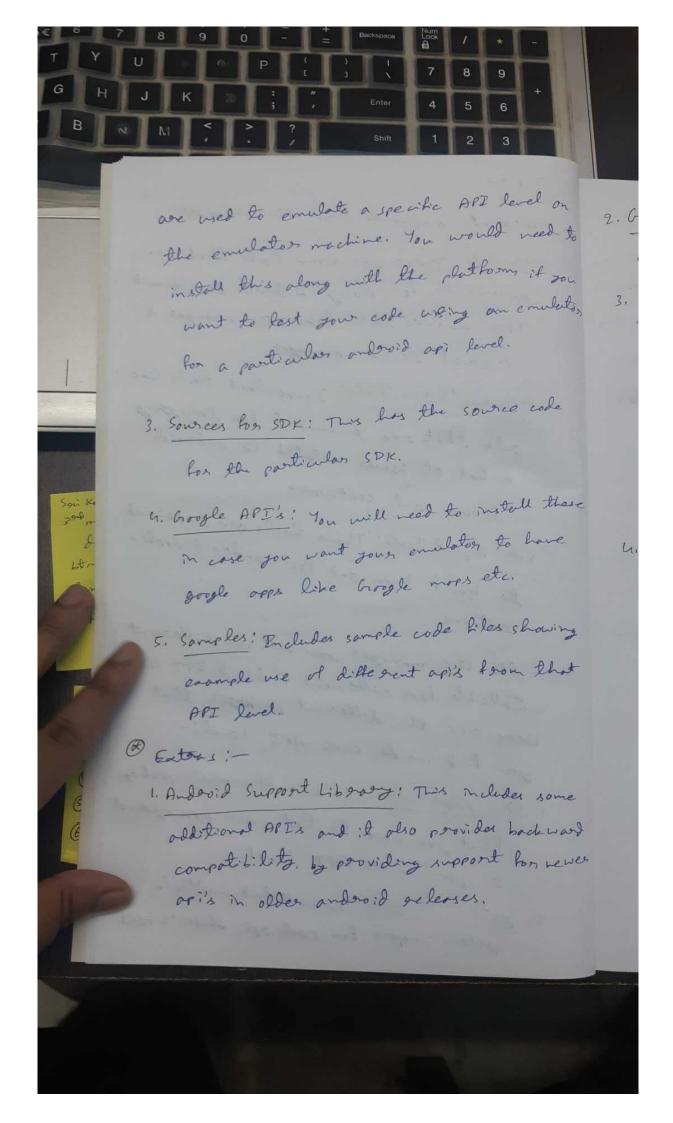
=) Native Apps:-They are specifically targeted to a morticular mobile platform in a specific language. Android apps - Kothy/Jan-Android studio IDE ios opps - Swift/Objective-C- xcode IDE 05. The development foramework provides mention Lion modimum feature access to the mobile platfogm and high performance. Advantages: Full feature access Performance. Disadvantage: High development cost, higher, learning curve. Orative opps can sun as standalone application.



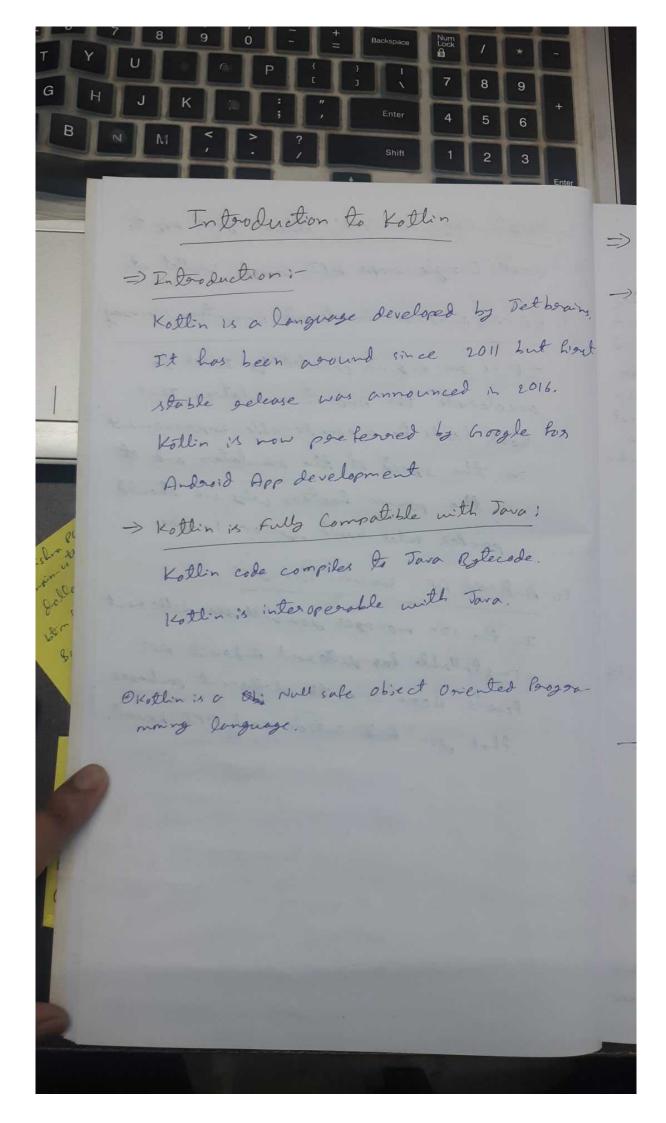
=> Coross-platform Apps: _ Coross-platform opps like Hybaid opps tobe advantage of single codebase. Hre woson difference is the wost they nender The lagort wing notive components, This beløs then parovide reas notive UX. Tools: Xamanin (C#), React Notive (Javascaipt), Flutter (Part) etc. etc. Advantage: Single codebase, Lower developm 大 ent cost Faster development. Notire UI look and feel, choice of development longuage. Disadvantage: Berlogmance, Ceature accessibilitz.



=) SDK :-@ Different Components from SDK Munager: -1. SDK Tools: These are some tools that are genised for doing Android development. This includes tools like SDX manges & Vigitual device manager. 2. SDK Platform Tools: Important Gools like adb that are required for debugging a lot of issues related to android devices and software. 3. SDK Build Tools: These Dools are required to build, our and test android opplications. OIn the SDK manager you will see different installable for different Android API levels. Here are the different packages that you find mi de each API Level:-1. SDK Plathogm! This is the main package That you need to test your opp against a particular API level. 2. Seter In oges: You will had multiple system images for each opi, these images



2. Grougle Play Services: Allows your upp to to access troogle wass API, Google Vallet etc. 04 3. HAXM (Hard wave Accelerated Execution Manager) itos It is an engine created by Intel to accelerate the Android emulation. This tool leads to a considerable improvement in the speed of the emulators and it is the paine lactor why we should parefer ntel based system images, 23-0 4. Android API specific Pochages: In the SDX manager you will see different mstallable for different Android API levels. Here are the different packages that you find inside each API Level.



-> Why Kotlin?; - Mulability: Good programming practice says that a out variable should change its value less often and preferably not at all. In Tava we have an optional "final" but in Kotlin its much more pobust. Ve reed to clarify the mutability of a variable in the declaration itself. mulabilita plays a very impostant part in multithoreaded code, where different codes want to change a samiable.

Null Safely: Extravables value is "null" means
one of the most common source of opp

one of the most common source of opp

cooshes during guntime is Null Pointes

exception. This happens when we too to

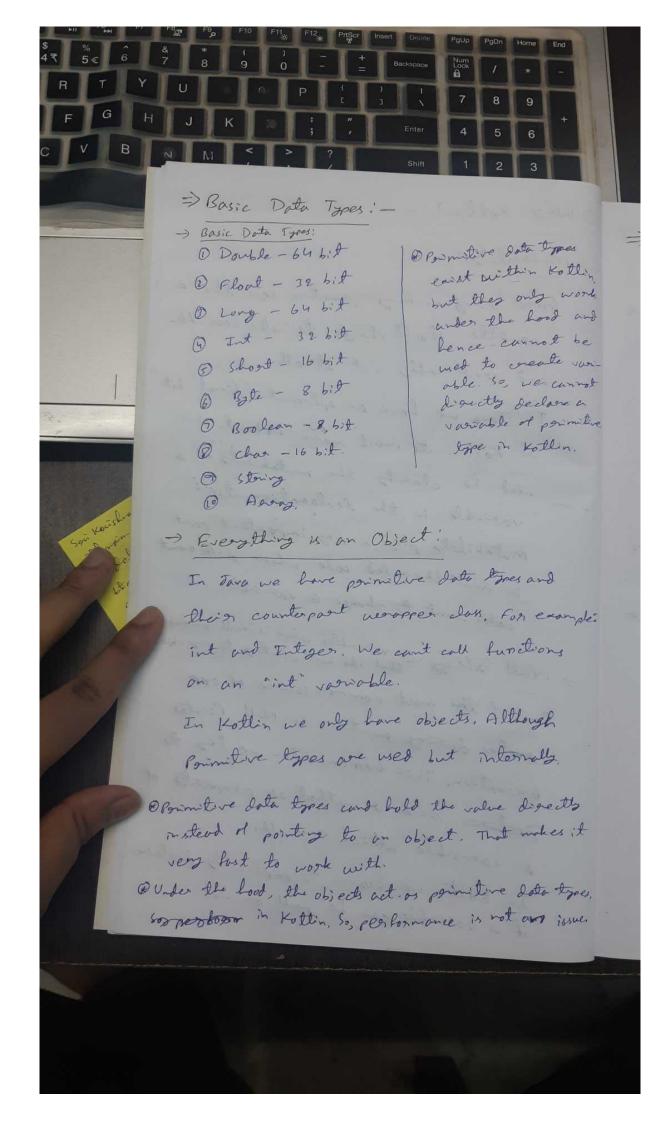
exception. This happens when we too of

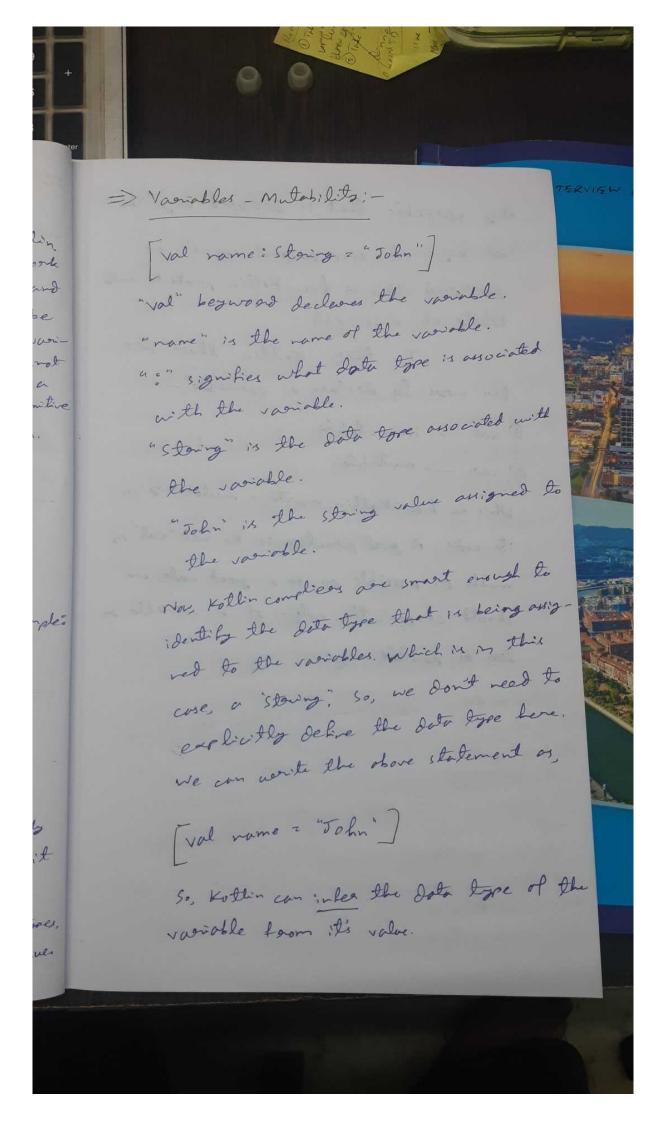
declare access a method or property of

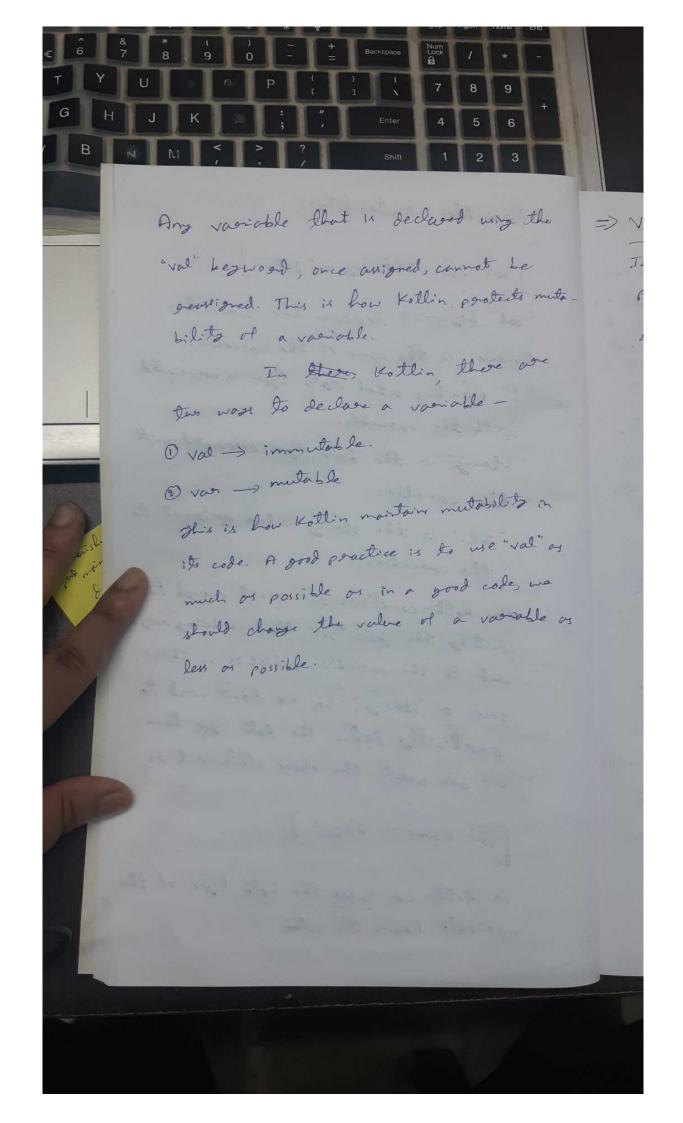
a variable which is null.

In Kotlin we have to explicitly declare

if a variable can be null on mot.







=> Nariables - Null Safety :-

If a variable holds a null value, i.e., it's pointing to no object in memory hear, for such a variable if we try to access it's function or property, we get a not pull Pointer Exception.

This Null Safety.

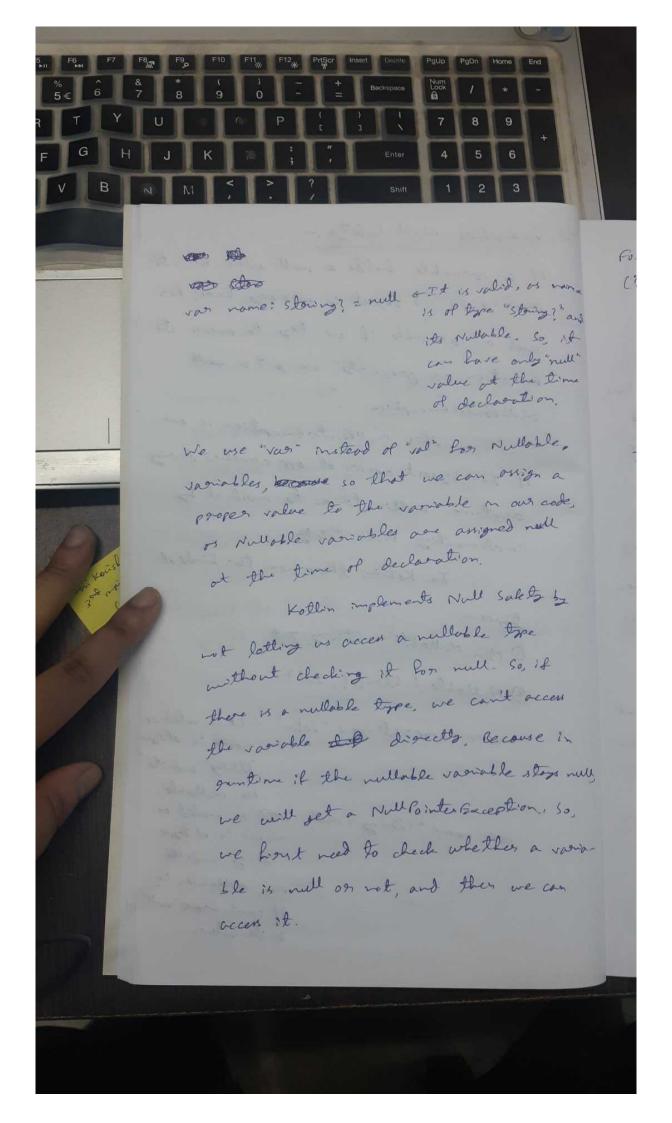
In Kothin, there are two kinds of

Dynes - O Non Mullable: Storing, Int

@ Nullable: Storing?, Int?

van name: stowing = "John" < It is valid of name is of type "stowing" and its Non Nullable.

vae name: stowing = null < It is invalid as name is of type "stowing" and its Non Nullable. So, it can't beave "null" as its value.

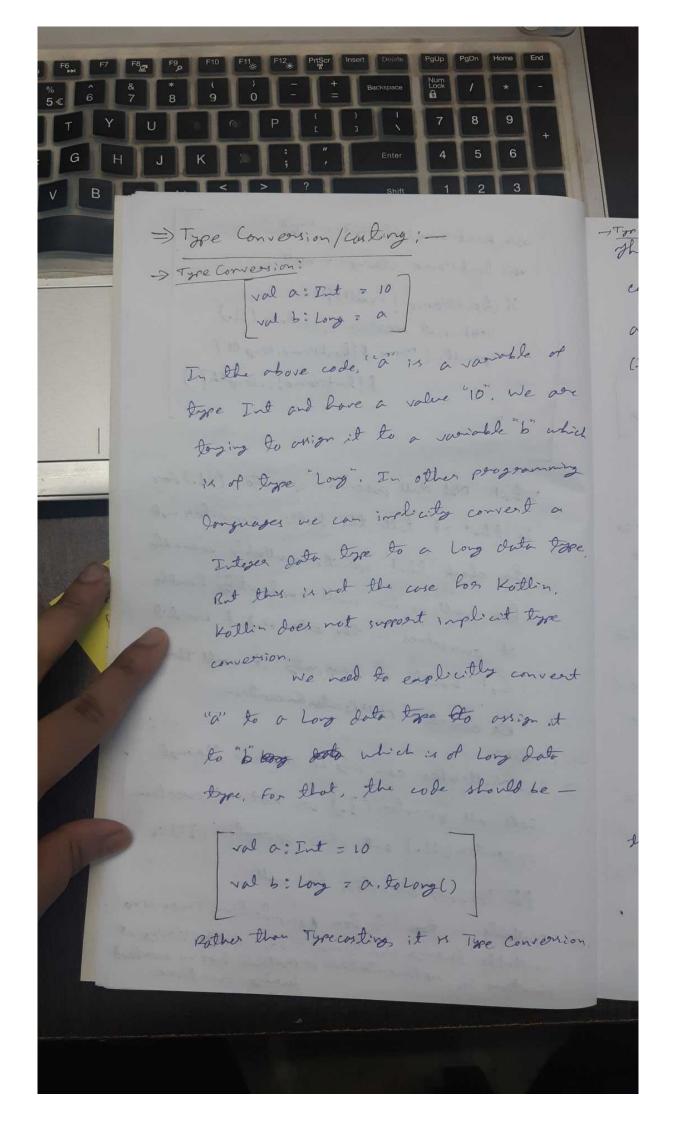


For null checks, Kotlin has a sale call operator, (?). For example :-3 and 11 De Jaration 11 var vame: Storing? = null ull" 11 safe call operatos ?. paint (name? length) In the above code, name is a mullable type. And when pointing the value of name's length, by accessing the "length" property of the sociable, we are putting the sale call operator (?) at the cut of anami So that Nullointer Exception is not thrown. what (?) does is that, it checks if the sociable is "null". It it's mull then it getugnes a "rull" value weithout occessiis it value on property and it it's not rull then it executes the code. U, For better toking care of the Wullowter Exception, we can use the Elvis Operator (?:) along with the Null Safety check Operator (?). For example.

11 Declaration von name; stowing? 2 nell 11 Sale Call Operator ? & Elvis operators ?: paint (name?.length ?:0) In the above code, name is a nullable tree If it we "name" is not then due to Sale Coll Operator, uname?.lugth" will geturn will and then due to Elvis Operator as the operand "rame? length" getigns well, "O" will be pointed. It "name? length" did not actuan will then their original value will be painted. So Elvis operator evaluates the operand for well. shere is another operator in context to rull. It is called "not rull ascertion operation (!!)". In Kothin only safe (?.) on not-nul asserted (!!.) calls are allowed on a nullable acceiver of type stowing For example:

The terror operator for rull.

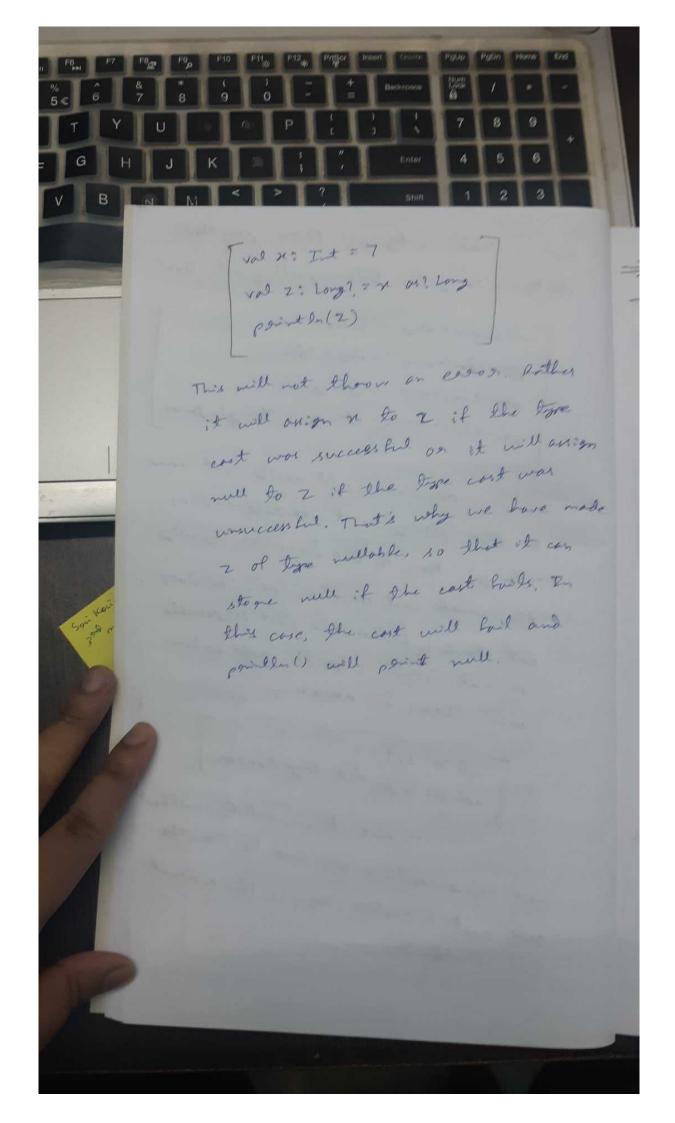
(2) Nullable is too only lon compile time. There is no mulloble during gruntime. There is only storing at guntime so, implementation of Nollable has no overlead during green time.



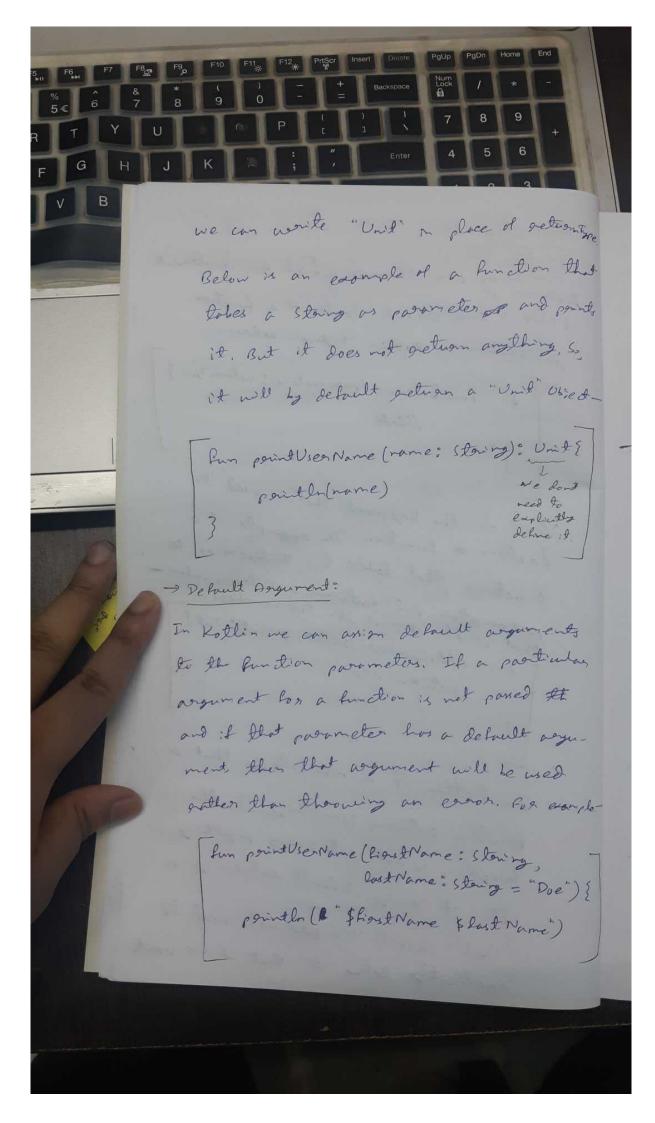
-styre Costing: There is a data true in Kotlin, which is colled "Any" 'Any" is a super class of all the closses. "Any" is the base class. (It's like the "Object" days of Kotlin. Just like, "Object" class is the Suger dass of all noc closses in Tora, "Any" is the Super class of all classes on Kotlin). So, any variable which having date type "Any" can hold are tree of object. (As everything in Kothin is an Object & there are no pointives). Tope We check the tree it a sociable ving the "15" operator, For course !at val name: Any = "John" pointln (name is stowng) The obove code will point "Tome" the value stor assigned to I "nome" is of type "storing" But "nome" itself is of Type "Any" that can hold my type of object.

Jamable To bresast a desect of "Any" type, we are the lectured to the type of the value it holds, we use "as" he grood. For example. val nome: And 2 " John" val name 1: Stan og 2 name ar Staning In the above code, "name" is of type "Any" but flowing value of type "storing" But in the second line, we convert "nome" to a "story" tree and origin it to "name 1" which is of tone "story" Using "05" doles not permanently convert "rune" to type "Stowng". It just sives us a convented version to assign to "rome", "rome" in still of type "Any" In the about example, we gould easity desila/ cost "frame" to troop "stoing" os it was But or "name" is assigned a "stowing" value it is interved or type "Storing" at compile

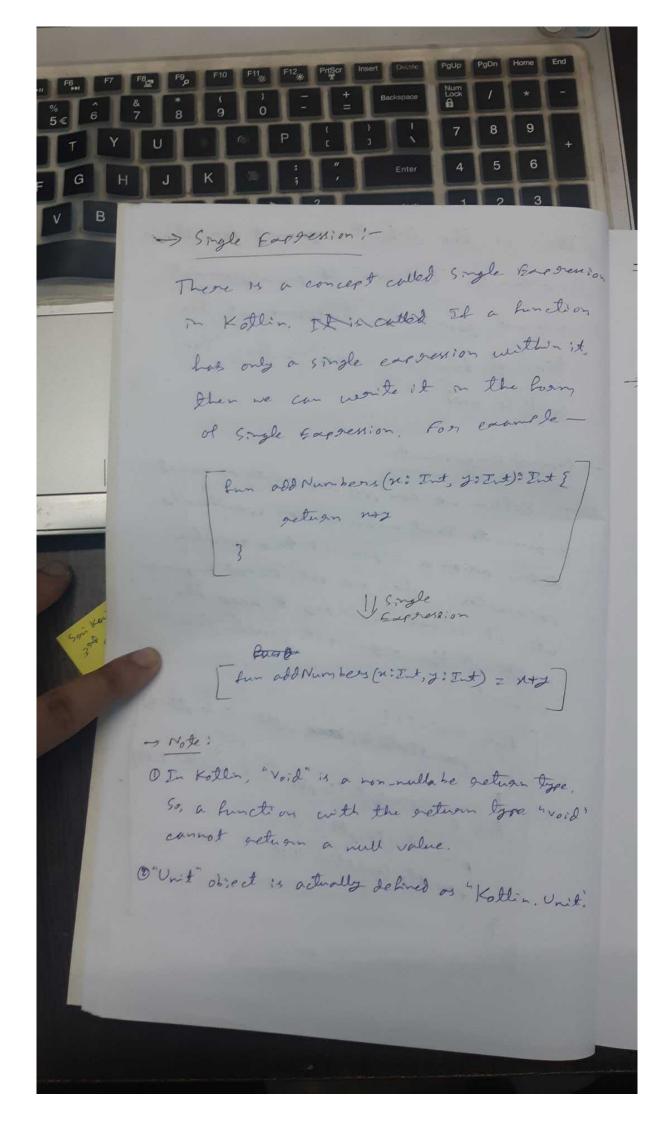
Jome Trut's why all those pointln() statements given below will point "tome". Tval name: Ang = "John" pointln (name is Any) 11 tome point In (name is Storing) 11 towe pointly (name as storing is storing) 11 touses Now, we could easily cest "name" to type "stoning", as it was inferred as "story" the by the consiler, by using y y the unsafe cost operator "as" loes not check whether the cost is possible on not That's why the code below 3 will those an error rt val n: Int = 7 val J: Long = n as Long //enron To cost it successfully without corrogs/exceptions, we need to use the safe cost operator (as?). So, the convect code should be



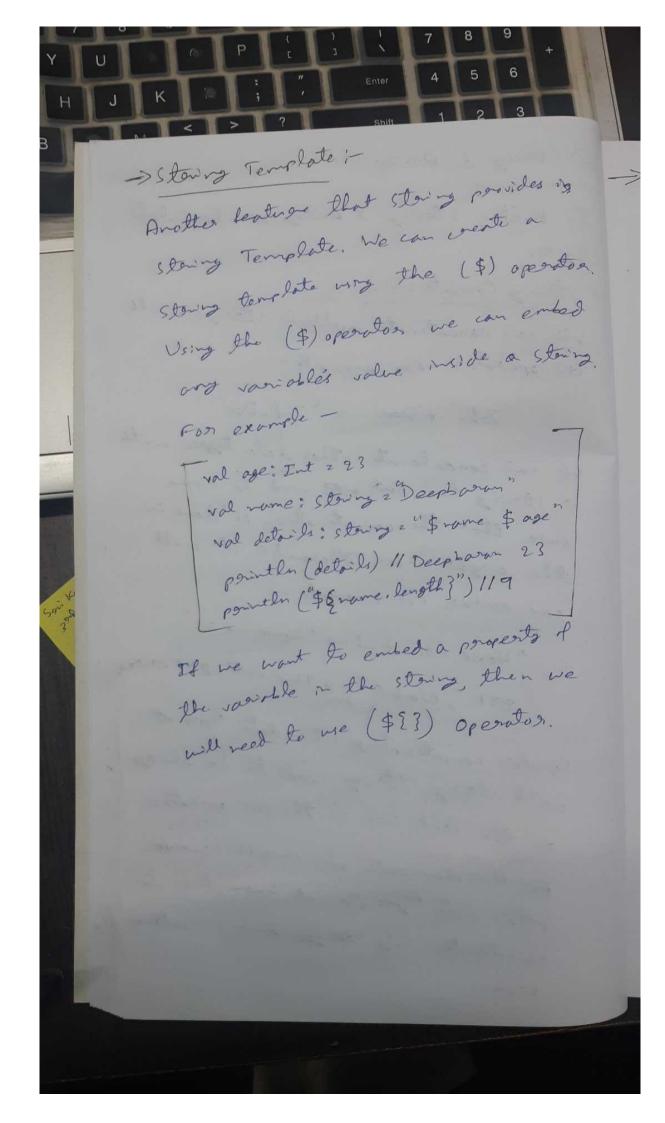
=> Functions: - Function Bosics: Functions are though which we implement leatures of I closes and objects. Syntax of a function de clasation on Kotlin is given belowfun functionName (parameters): getvanType { 11 Code de "fun" is the keywood that is used to de clare a function. An example of a function that takes two Integers as its parameter and geturns a Integer value. Hot is the sum of the two numbers fun add Numbers (x: Int, y: Int): Tut { getuan n+7 In Kothin, it is mandatogy that a function getugny something, If the function is not getnaming anything then it is by defoult getnoming a "Unit" object. We do not need to earlicitly define it. But if we want,



"Doe" is the default argument for the paranetes "Past Nome". It an argument Dy was not passed for "DastName" at the time of function invokation, then "Doe" will be used. - Named Agguments: In Kotlin we can pass arguments as named argunests. That means we can specifically define onion a argument to a specific parameter. when we we warred Argune nts, it is not necessary to heer the order of the organients possed. For example. point Delails (width = 10, length = 5) for main () { Pun point Details (length: Int, width: Int); pointly (length) 115 pointly (width) // 10



=> Stowng & Agray:milos In Kolting Storing & Assay are basic data X trees. -> Storing Concatenation: We can concatenate multiple storys with (+) operator. For example-"John" + Doe" - Tohn Doe" We can concaterate other data types with a stowing two, But we reed to make sure that the concatenation is done in the night order, Otherwise it will gre error For example " House no," +23 -> "House no, 23" 23+", Block (" -) This will give error, So, for concatenating other data trees of with storing, storing reads to be always on the left side of the (+) operator. En the oboxe coder to This is because. other date types do not know how to conceterate. Only storys know concetera-



-> Aggaz: Agrage are zero indexed in Kotlin (Agray) in Nothin can have multiple data types.) creating array in Kothin: val numbers = acraz Of(1,2,3,4)]
val numbers = acraz Of(Int)(1,2,3,4)] (This works but we should noton from doing it Val numbers 2 z intagory Of (1,2,3,4) Getting data from array using index: numbers. get(1) 1/2 (T1) - Index operator numbers [2] 113 Changing values of array using whea? numbers, set (3,5) numbers [1] = 22 Painting the away: pountle (numbers, jointo storing (",")) 1/1,22,3,5 Coesting a covery of specific site and fill it we with a particular value: val numbers Value = Agray (3) {5} //[5, 5, 5] This @ This is how we can initialize Lower is created, on good of a specific site. Size of an agray: numbers, size 1/4

В -> Note: O storings are immutable in Kotlin. @ Accessing storing indexel! sta, get (mdex) stor [inde a] 3 multidemensional Arroy: val and = acroyof (1, 2, 3) val arre 2 arrayof (4, 5, 6) val and 2 around of (onest ares) val areas = Arras (2) {Array (3) {0}} [[000]] @ Cocating on enote stoning object to hall geter: var s z staing U @ coesting a storing object: your s = stoning ("Hello") 6 substant. val s z Deepbaran pointln (5, subsequence (0,2)) 11 De

a Storing Companison:

val S1 = "Deep" val 52 2 " Deep"

real 53 2 "barran"

51. eanals (52) 11 tome

\$1. compare To (52) 110 veryod. 52, compare To (53) 11 And numerical value.

- DAS Kotlin gurs on JVM, so it uses the same gambage Collection as Jan on any other JVM Losed language. So, Kother uses Garbage collection.
- @ Kother stores only references to our objects into stack memory. The objects are actually stored in the hear memory.
- 10 In Kotlin, everything is an object (reference type, not pointive type), We don't bond powmitive types, like the ones we can use in Java, This greduces code complexity, we can call nethods and persperties on any variable. For evaruple, this is how we can convert the Int variable to a char. Usually curhenever it is poss-Tale), under the hood types such or Int, Long, on char are optimized (stored as point the types) but we can still call methods on them as on any other objects.

