Q) Tell me why a class without any constructor but properties need to be initialized while a class having primary constructor need not to initialize its properties at time of class definition in Kotlin

In Kotlin, this behavior is due to the distinction between primary constructors, secondary constructors, and how properties are initialized in a class.

Here's a breakdown:

1. Class Without a Constructor

If a class does not have any constructor, but has properties, you need to initialize those properties **explicitly** in one of the following ways:

- By giving the property a default value in the class definition.
- By initializing them inside an initializer block (init block) or a secondary constructor.

Example: Way-1

```
class Example {
   var name: String // Property without initialization

   init {
      name = "John Doe"  // Initialized inside the init block
   }
}
```

Way-2

```
Class Example {
        Var name: String =“” //default value to initialize object
        Var id: Int = 0
}
```

Main---

```
var obj= Example()
println(obj.name) // “”
println(obj.id) //0
obj.name= “Rohan” // we can change these default value explicitely
```

Since no primary constructor is defined, Kotlin expects you to explicitly initialize all properties. If you do not provide a default value or initialize them in an init block, you'll encounter an error saying the property is not initialized.

2. Class With a Primary Constructor

In contrast, when a class has a **primary constructor**, you can initialize the properties directly within the constructor's parameter list. This allows for concise property initialization and avoids the need for an explicit initializer block.

Example:

```
class Example(val name: String)  // Initialized directly via primary constructor
```

In this case, the name property is initialized by the constructor parameter when an object is instantiated. Kotlin handles the initialization automatically when the constructor is called, so there's no need to explicitly initialize the property elsewhere.

Why the Difference?

- **Without a constructor:** Kotlin doesn't know when or how to initialize properties unless you explicitly do it in the init block or set a default value.
- **With a primary constructor:** Kotlin knows that the properties will be initialized when the class is instantiated with arguments passed to the constructor. The constructor parameters provide an initialization mechanism that removes the need for an init block or manual assignment.

Summary:

- **Class without a constructor:** You need to initialize properties manually (e.g., in the init block).
- **Class with a primary constructor:** Properties can be initialized via constructor parameters, so no additional initialization is needed.

```
Class Example (n: String, i: Int){

Val name= n // default value

Val id= i

Val smell : String // no error as value get initialized in init block

        Init{

                Smell= “good”

        }

}


Fun main(){

        Val ob= Example()

        Println(ob.name)

        Println(ob.id)

        Println(ob.smell)

}
```