Q) variable in primary constructor can be called as properties of not, or there is only case variable defined in class body is called as properties

In Kotlin, variables defined in the **primary constructor** can indeed be **properties**, but only if you explicitly declare them with var or val. If you don't declare them with these keywords, they are treated as constructor parameters, not class properties.

Here's a detailed explanation of both scenarios:

1. Primary Constructor Variables as Properties

If you define a variable in the primary constructor with var or val, it **becomes a property** of the class. This means it is accessible throughout the class and can be used just like any other class property (with either getter/setter for var or a getter for val).

Example:

```
class Person(val name: String, var age: Int)
```

Here:

- name is a **read-only** property because it's declared with val.
- age is a **mutable** property because it's declared with var.

Both name and age are properties of the Person class, accessible throughout the class and via instances of the class:

```
val person = Person("John", 25)

println(person.name)  // Accesses the property 'name'

person.age = 26       // Mutates the property 'age'
```

2. Primary Constructor Variables Not as Properties

If you do **not** use var or val in the primary constructor, the variables are just **constructor parameters**. They are not properties of the class, meaning they are only available within the constructor (or init block) and cannot be accessed later outside of those.

Example:

```
class Person(name: String, age: Int) {
    // These are constructor parameters, not properties
    init {
        println("Person's name is $name and age is $age")  // Accessible here
    }
}
```

In this case:

- name and age are **constructor parameters**, not class properties.
- You cannot access name or age outside the constructor or init block, for example:

```
val person = Person("John", 25)

// println(person.name) // Error: Unresolved reference 'name'
```

3. Properties Defined in the Class Body

Alternatively, properties can also be declared in the class body, outside of the constructor. These properties must be explicitly initialized or assigned default values.

Example:

kotlin

Copy code

```
class Person {
    var name: String = "Unknown"  // Property in the class body
    var age: Int = 0              // Property in the class body
}
```

These properties are accessible and mutable throughout the class:

```
val person = Person()
println(person.name)  // Outputs "Unknown"
person.age = 30       // Updates the age property
```

Summary:

- **Primary constructor parameters** declared with var or val are **class properties**.
- **Primary constructor parameters** without var or val are **not class properties**; they are just parameters.
- **Properties in the class body** are always properties, whether or not a constructor is defined.