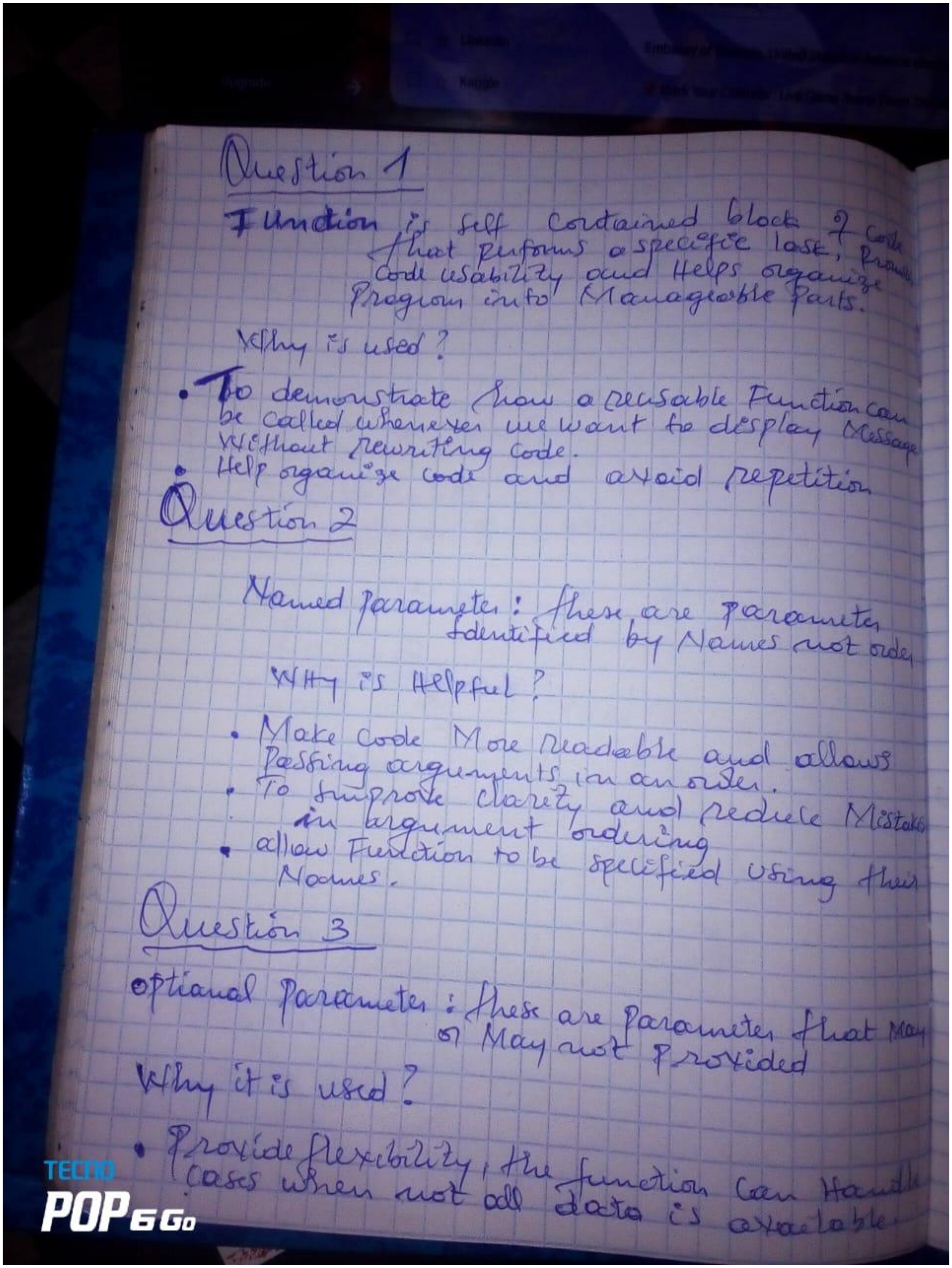


Question 1

Function is self contained block of Code that performs a specific task, Promo Code usability and Helps organize Program into Manageable Parts.

Why is used?

- To demonstrate how a reusable Function can be called whenever we want to display Message without rewriting code.
- Help organize code and avoid repetition

Question 2

Named parameter: these are parameter identified by Names not order

Why is Helpful?

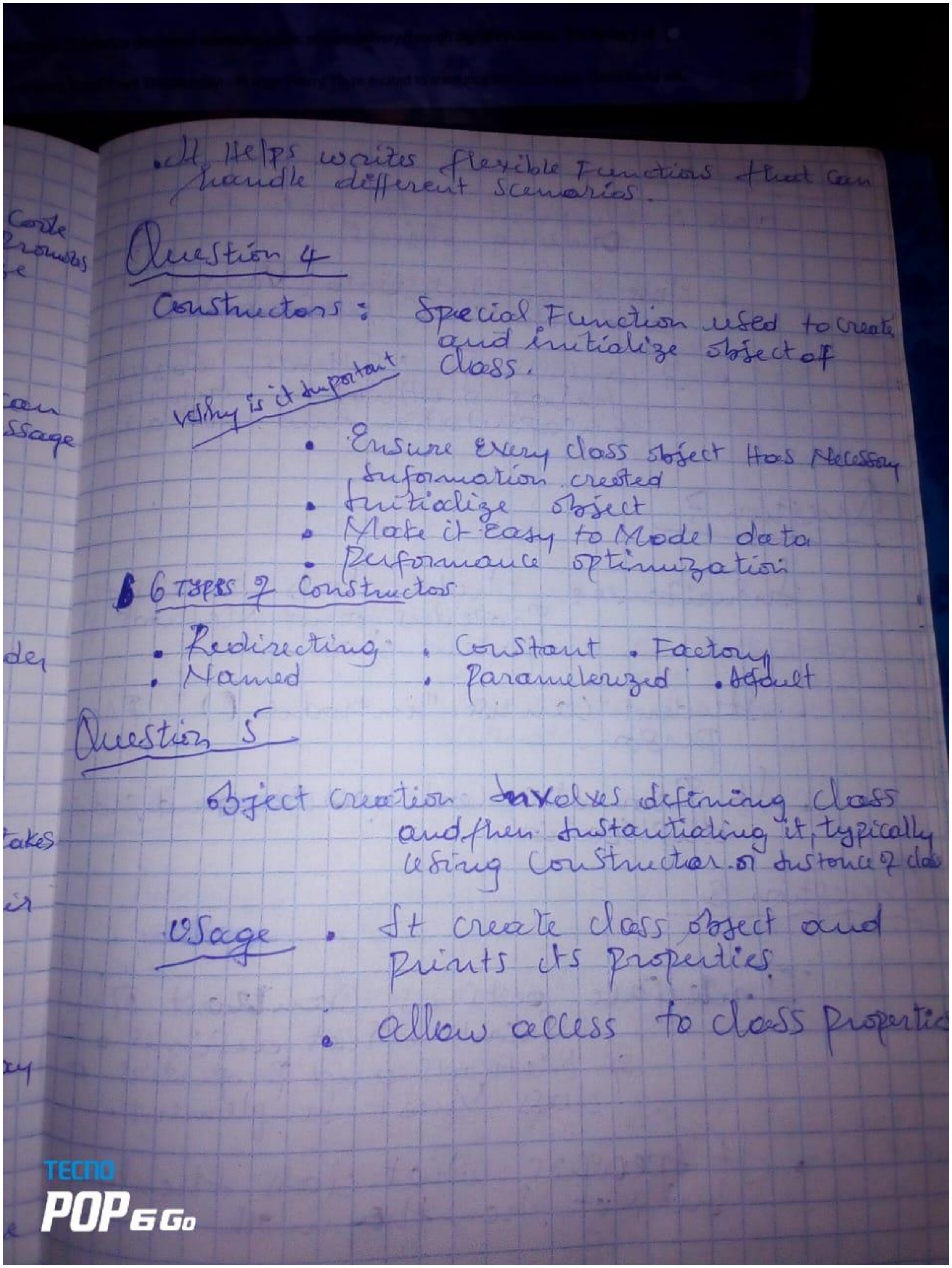
- Make Code More readable and allows Passing arguments in an order.
- To improve clarity and reduce Mistakes in argument ordering
- allow Function to be specified using their Names.

Question 3

optional Parameter: these are Parameter that May or May not Provided

Why it is used?

- Provide flexibility, the function can Handle cases when not all data is available.



Code
Promotes
e

can
ssage

der

takes

ir

ay

e

- It Helps writes flexible Functions that can handle different Scenarios.

Question 4

Constructors : Special Function used to create and initialize object of Class.

Why is it important

- Ensure every class object Has Necessary information created
- Initialize object
- Make it Easy to Model data
- Performance optimization

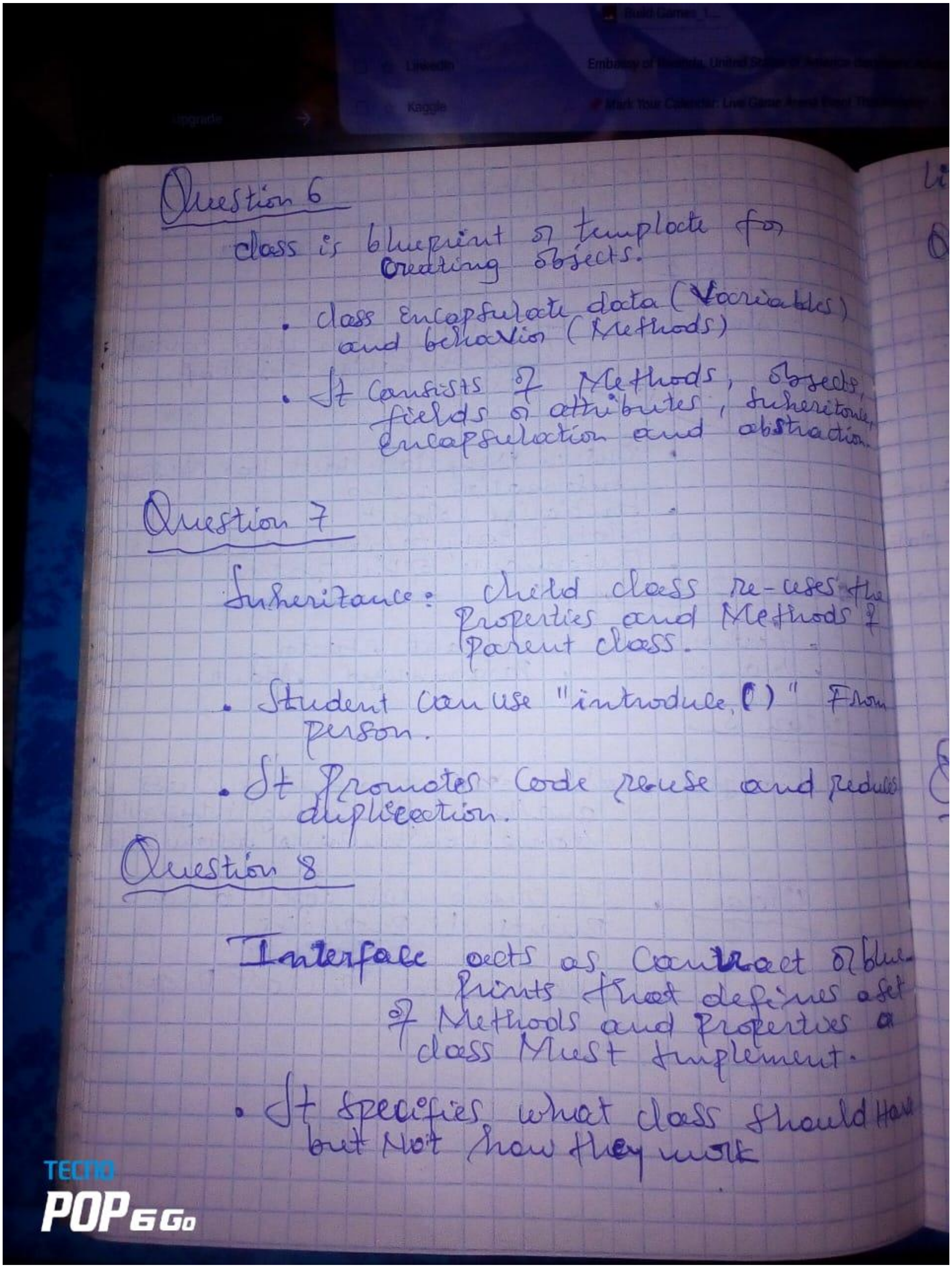
6 Types of Constructor

- Redirecting
- Named
- Constant
- Parameterized
- Factory
- default

Question 5

Object creation involves defining class and then instantiating it, typically using Constructor or instance of class

Usage
- It create class object and prints its properties
- allow access to class propertie



Question 6

class is blueprint or template for creating objects.

- class Encapsulate data (Variables) and behavior (Methods)
- It consists of Methods, objects, fields or attributes, Inheritance, Encapsulation and abstraction.

Question 7

Inheritance: child class re-uses the Properties and Methods of parent class.

- Student can use "introduce()" From person.
- It Promotes Code reuse and reduces duplication.

Question 8

Interface acts as Contract or blue-prints that defines a set of Methods and Properties a class Must implement.

- It specifies what class should Have but Not how they work like : "Registrable" acts as an interface

Question 9

- Implementing Interfaces

  - Use "Implements" Keyword
  - Interface can be any class, bcz in dart every class defines an interface.
  - Each class Must provide concrete Implementation of all Methods in the interface.

- Enforcing rules

> Method Existence
> Consistency (
> ~~use "Extend" keyword for Inheritance~~
> Safety (code that works with Interface without worrying about Missing Methods.

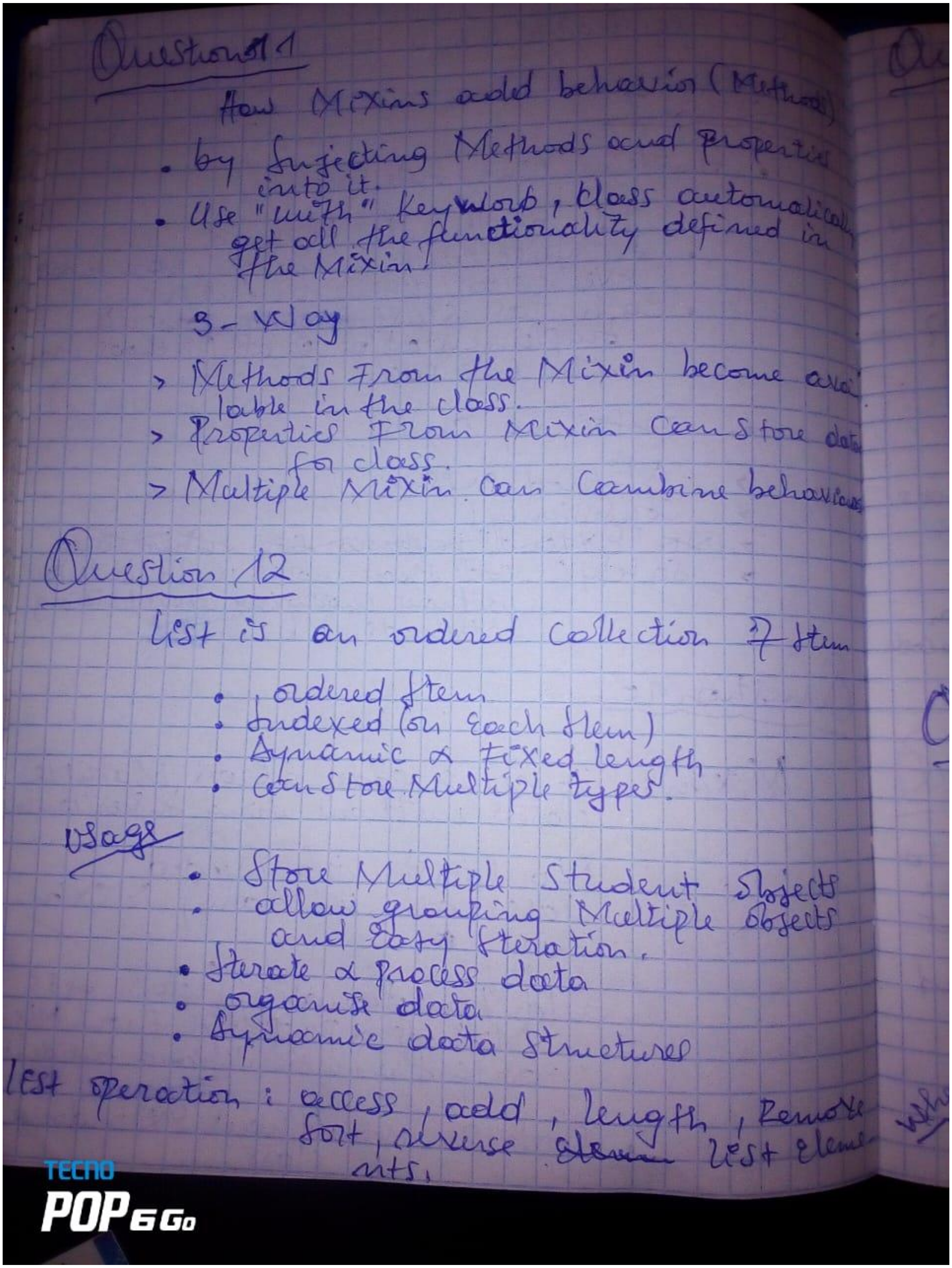
Question 10

Mixin : to re-use code across Multiple classes without using Inheritance.

- No Constructor Needed
- class can use Many Mixin at once.

- adds Functionality (class gets Methods & properties From Mixin as if they were defined Inside it.

- Code re-use without Extending class.



Question11

How Mixins add behavior (Method)

- by Injecting Methods and properties into it.
- Use "with" Keyword, class automatically get all the functionality defined in the Mixin.

3 - Way

> Methods From the Mixin become available in the class.
> Properties From Mixin can store data for class.
> Multiple Mixin can combine behaviours

Question 12

List is an ordered collection of Item

- ordered Item
- Indexed (on Each Item)
- dynamic & Fixed length
- can Store Multiple types.

Usage

- Store Multiple Student objects
- allow grouping Multiple objects and Easy Iteration.
- Iterate & process data
- organise data
- dynamic data structures

List operation : access, add, length, Remove sort, reverse ~~Stem~~ list Elements,

Question 13

Map is collection of key-Value pairs where Each key is unique.

- Flexible data types
- Dynamic size
- No Direct Indexing

- Unique keys
- Access by keys

Basic operations

- add / update
- Retrieve
- Remove
- Check Existence

It is useful When:

1. Quick Lookup (find data by key Instead of looping through list
2. Storing J-son-like data: (is used to repre-sents Json objects)
3. Counting & grouping
4. organizing data.

Question 14

Anonymous Function is Function that doesn't have Name, usually Inline to pass arguments for small and temporary tasks.

- Are used when you don't Need to re-use the Function Elsewhere.

When to use

- Short lived Functions
- Pass Function as arguments
- Cleaner Code (keeps small logic inline)
- Closures (Capture Variable From Surrounding

. It can be . { > AnonyMous Function as param
> Arrow Syntax.

Question 15

Arrow Function is Short and Simplified way to write a function that contains only one expression.

. It use "=>" Symbol instead of curly braces.

Why arrow Function Improve Simplicity

. less Code . More redeable
. Clear intent . perfect for callbacks.

When to use Arrow Function

- Function Has only one expression
- you want simple, clean & readable code
- you Need Multiple Statements
- you Need Complex logic or Loops

Question 16

Asynchronous Functions allow Program to perform time Consuming tasks (like loading data, reading file) without blocking the rest of the program.

. It use async, Future, Wait as keyword

Why Ans Function are important.

- Non blocking Execution
- Better performance
- Essential for real apps

"Waiting" Means Pausing Execution at specific point until an Asynchronous operation Completes, without freezing the whole Program.

- Use "Wait" keyword

Question 17

Problem Async solves in Real apps

> Fetching data From internet (API)
> Read/writing files
> Accessing databases
> waiting for timers or user actions

It Helps To:

- Keeps APP responsive (No Freezing)
- Improves User Experience
- Enables Background work
- Handle Multiple Tasks Efficiently
- Essential for Networking & API

Question 18

Why Mixins are useful

- Code re-use without Inheritance
- classes gain specific behaviour
- you can apply Multiple Mixin to the Same class
- they keep designs Flexible and Modular

BTN Inheritance & Mixin

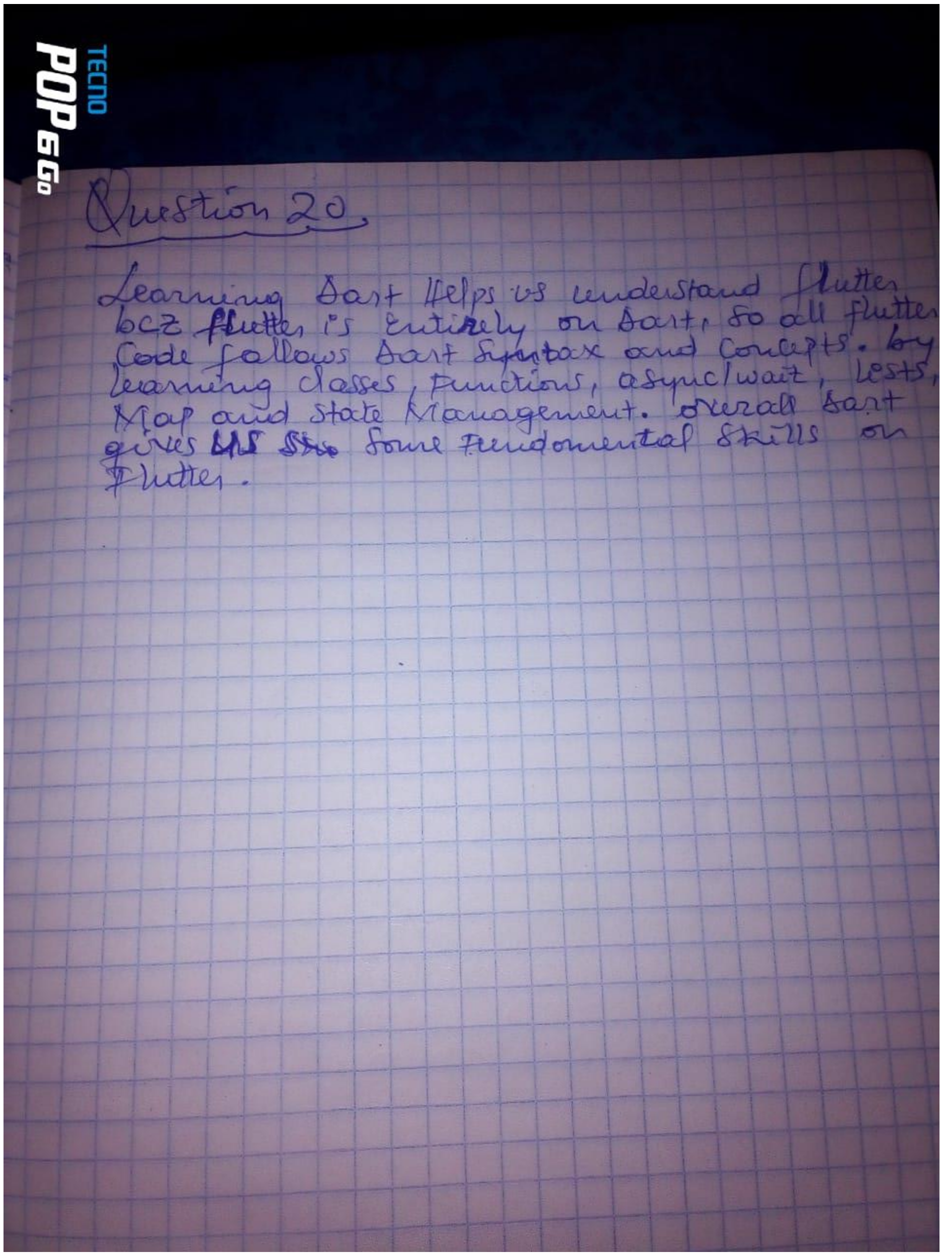
Aspect	Inheritance	Mixins
Purpose	Model "relationship	. Shared behaviour /
Example	. Student is person	. Student can Have attendance
Nº allowed	. one Superclass only	. Multiple Mixins allowed
Use case	. Core Identity	. Optional or re-usable features
Constructor	Subclass Inherit	. do not Have Constructor

Question 19

How New Mixin was used?

1. Mixin was created to contain specific behaviour
   Ex: NotificationMixin was designed to Handle Notification when Student register for course.
2. The Mixin was applied to the class
3. The Mixins Method was called From objects

⚠ New Mixin is used to add Extra behaviour to the class without changing its Inheritance, Making code clean, reusable and Easier to Extend later.



Question 20.

Learning Dart Helps us understand flutter bcz flutter is Entirely on Dart, so all flutter Code follows Dart Syntax and Concepts. by learning classes, Functions, async/wait, Lists, Map and State Management. overall Dart gives ~~us~~ some Fundomental Skills on Flutter.