🧠 SET A – Coding Practice (Detailed)

🔹 Q1 (Basic – Kotlin Lambdas & Higher Order Functions)

📝 Problem Statement:

Create a Kotlin program where:

You define a higher-order function named calculate.

It should take:

Two integers

One lambda function as a parameter

The lambda should perform a mathematical operation.

Call the function for:

Addition

Multiplication

📌 Requirements:

1. Create function:

fun calculate(a: Int, b: Int, operation: (Int, Int) -> Int): Int

2. Call it using:

Lambda for addition

Lambda for multiplication

3. Print the results.

⭐ Expected Output:

Addition Result: 15

Multiplication Result: 50

🧠 Examiner Checks:

Correct lambda syntax

Understanding of higher-order functions

Correct function return type

🔹 Q2 (Medium – Coroutines Network Simulation)

📝 Problem Statement:

Create an Android activity that:

1. Shows a button "Fetch Data"

2. When clicked:

Start coroutine

Switch to IO dispatcher

Simulate network call using delay(3000)

Return result to Main thread

Display result in TextView

📌 Requirements:

Use CoroutineScope

Use Dispatchers.IO

Use withContext(Dispatchers.Main)

Use delay()

Show loading message before delay

⭐ Expected Flow:

1. Click button

2. TextView → "Loading..."

3. Wait 3 seconds

4. TextView → "Data Fetched Successfully"

🔒 Constraints:

Do NOT use AsyncTask

Must use Coroutines

🧠 Examiner Checks:

Correct dispatcher usage

Thread switching knowledge

Coroutine builder understanding

🧠 SET B – Coding Practice (Detailed)

🔹 Q1 (Basic – Scope Functions)

📝 Problem Statement:

Create a User data class with:

name

age

city

Use apply scope function to:

Create object

Initialize values

Print details using also

📌 Requirements:

1. Define data class

2. Use:

val user = User().apply { ... }

3. Print using:

user.also { println(it) }

⭐ Expected Output:

User(name=Viney, age=22, city=Delhi)

🧠 Examiner Checks:

Correct use of this in apply

Understanding difference between apply & also

🔹 Q2 (Medium – Bound Service)

📝 Problem Statement:

Create:

One Activity

One Bound Service

Service should:

Generate random number

Return it to Activity

Activity should:

Bind to service

Call service method

Display number in TextView

📌 Requirements:

Override:

onBind()

Create inner Binder class

Use:

bindService()

⭐ Expected Flow:

1. Activity starts

2. Service binds

3. Button click → show random number

🔒 Constraints:

Must use Binder

No Intent extra allowed

🧠 Examiner Checks:

Proper binding lifecycle

IBinder understanding

Service connection handling

🧠 SET C – Coding Practice (Detailed)

🔹 Q1 (Basic – Extension Function)

📝 Problem Statement:

Create an extension function for String:

Function name: isPalindrome()

Return true if string is palindrome

Ignore case sensitivity

📌 Requirements:

1. Write:

fun String.isPalindrome(): Boolean

2. Test it with:

"madam"

"Hello"

⭐ Expected Output:

madam -> true

Hello -> false

🧠 Examiner Checks:

Correct extension syntax

String manipulation logic

🔹 Q2 (Medium – Custom Content Provider)

📝 Problem Statement:

Create a custom Content Provider that:

Supports:

Insert

Query

Data stored in:

SQLite database

Another activity should:

Insert student record

Fetch all records

Display in Logcat

📌 Requirements:

1. Extend ContentProvider

2. Override:

onCreate()

insert()

query()

3. Define URI

4. Register provider in Manifest

⭐ Expected Flow:

1. Insert student

2. Query returns all students

3. Logcat shows results

🔒 Constraints:

Must use ContentResolver

Must define authority in Manifest

🧠 Examiner Checks:

URI handling

CRUD understanding

IPC concept clarity