✅ SET A

📝 SET A – Coding Question 1 (Basic)

❓ Question:

Create a higher-order function calculate() that takes two integers and a lambda operation. Perform addition and multiplication using lambdas.

✅ Answer:

```
fun calculate(a: Int, b: Int, operation: (Int, Int) -> Int): Int {
    return operation(a, b)
}

fun main() {

    val addition = calculate(10, 5) { x, y -> x + y }
    val multiplication = calculate(10, 5) { x, y -> x * y }

    println("Addition Result: $addition")
    println("Multiplication Result: $multiplication")
}
```

📝 SET A – Coding Question 2 (Medium)

❓ Question:

Create an Android app that:

On button click

Uses coroutine

Simulates network call using delay

Shows result in TextView

✅ Answer (MainActivity.kt):

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import kotlinx.coroutines.*
```

```
class MainActivity : AppCompatActivity() {

    private lateinit var textView: TextView
    private lateinit var button: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        textView = findViewById(R.id.textView)
        button = findViewById(R.id.button)

        button.setOnClickListener {

            textView.text = "Loading..."

            CoroutineScope(Dispatchers.IO).launch {

                delay(3000)

                withContext(Dispatchers.Main) {
                    textView.text = "Data Fetched Successfully"
                }
            }
        }
    }
}
```

📚 SET A – MCQs (With Answers)

1. Lambda is:

B ✅ Anonymous function

2. Higher order function:

B ✅ Function taking function as parameter

3. Extension function used to:

B ✅ Add functionality without modifying class

4. Scope function returning object:

C ✅ also

5. let uses:

B ✅ it

6. Coroutine builder:

B ✅ launch

7. IO dispatcher used for:

C ✅ Network/Database

8. Flow is:

B ✅ Async stream

9. StateFlow holds:

B ✅ Latest value

10. Explicit Intent requires:

B ✅ Component name

11. First lifecycle method of Service:

B ✅ onCreate

12. Stop service using:

B ✅ stopSelf

13. AIDL used for:

B ✅ Cross-process communication

14. Content Provider performs:

B ✅ CRUD

15. JNI stands for:

B ✅ Java Native Interface

16. Native library extension:

C ✅ .so

17. NDK used for:

B ✅ C/C++

18. Frida used for:

B ✅ Hooking native methods

19. Messenger uses:

B ✅ Handler

20. async returns:

A ✅ Deferred

🧠 ==========

✅ SET B

==========

📝 SET B – Coding Question 1 (Basic)

❓ Question:

Create a User class and initialize using apply. Print using also.

✅ Answer:

```
data class User(var name: String = "", var age: Int = 0, var city: String = "")

fun main() {

    val user = User().apply {
        name = "Viney"
        age = 22
        city = "Delhi"
    }

    user.also {
        println(it)
    }
}
```

📝 SET B – Coding Question 2 (Medium – Bound Service)

✅ Service Class

```
class MyService : Service() {
```

```
    private val binder = MyBinder()

    inner class MyBinder : Binder() {
        fun getRandomNumber(): Int {
            return (1..100).random()
        }
    }

    override fun onBind(intent: Intent): IBinder {
        return binder
    }
}
```

✅ Activity

```
class MainActivity : AppCompatActivity() {

    private var myBinder: MyService.MyBinder? = null

    private val serviceConnection = object : ServiceConnection {
        override fun onServiceConnected(name: ComponentName?, service: IBinder?) {
            myBinder = service as MyService.MyBinder
        }
```

```
        override fun onServiceDisconnected(name: ComponentName?) {
            myBinder = null
        }
    }

    override fun onStart() {
        super.onStart()
        Intent(this, MyService::class.java).also {
            bindService(it, serviceConnection, Context.BIND_AUTO_CREATE)
        }
    }

    fun showNumber() {
        val number = myBinder?.getRandomNumber()
        println("Random Number: $number")
    }
}
```

📚 SET B – MCQs Answers

1. apply returns → B ✅ Object

2. run returns → B ✅ Block result

3. with is → B ✅ Normal function

4. GlobalScope → B ✅ Not lifecycle aware

5. suspend called from → B ✅ Coroutine

6. Default dispatcher → A ✅ CPU

7. Flow cold because → B ✅ Starts when collected

8. putExtra used for → B ✅ Data sharing

9. Implicit Intent uses → B ✅ Action

10. Bound service ends when → B ✅ All unbind

11. Binder used in → B ✅ Bound service

12. AIDL needed for → B ✅ Different process

13. ContentResolver works with → B ✅ ContentProvider

14. JNI calls → B ✅ C/C++

15. .so is → B ✅ Shared object

16. Frida used for → A ✅ Security testing

17. onBind returns → B ✅ IBinder

18. Messenger alternative to → A ✅ AIDL

19. Coroutine scope controls → A ✅ Lifecycle

20. filter{} is → B ✅ Higher order function

🧠 ==========

✅ SET C

==========

📝 SET C – Coding Question 1 (Basic – Extension)

✅ Answer:

```
fun String.isPalindrome(): Boolean {
    return this.lowercase() == this.lowercase().reversed()
}

fun main() {

    val word1 = "madam"
```

```
    val word2 = "Hello"

    println("$word1 -> ${word1.isPalindrome()}")
    println("$word2 -> ${word2.isPalindrome()}")
}
```

📝 SET C – Coding Question 2 (Medium – Content Provider)

✅ Content Provider Skeleton

```
class StudentProvider : ContentProvider() {

    override fun onCreate(): Boolean {
        return true
    }

    override fun insert(uri: Uri, values: ContentValues?): Uri? {
        println("Data Inserted")
        return uri
    }

    override fun query(
```

```
        uri: Uri,
        projection: Array<out String>?,
        selection: String?,
        selectionArgs: Array<out String>?,
        sortOrder: String?
    ): Cursor? {

        println("Data Queried")
        return null
    }

    override fun delete(uri: Uri, selection: String?, selectionArgs: Array<out String>?): Int =
0

    override fun update(uri: Uri, values: ContentValues?, selection: String?,
selectionArgs: Array<out String>?): Int = 0

    override fun getType(uri: Uri): String? = null
}
```

📚 SET C – MCQ Answers

1. A ✅

2. B ✅

3. B ✅

4. A ✅

5. B ✅

6. B ✅

7. B ✅

8. B ✅

9. A ✅

10. B ✅

11. A ✅

12. A ✅

13. A ✅

14. A ✅

15. B ✅

16. B ✅

17. A ✅

18. A ✅

19. A ✅

20. A ✅