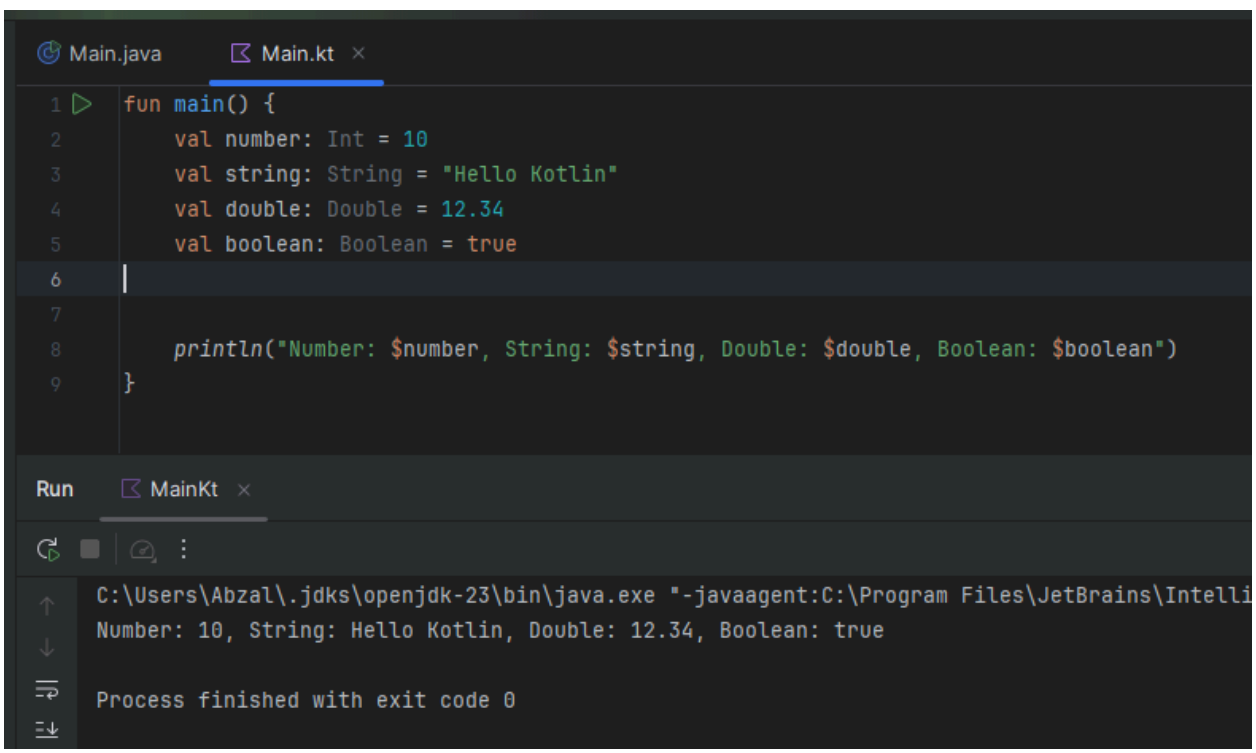
Assignment 1, Mobile Programming

Put all deliverables into github repository in your profile. Share link to google form 24 hours before defense. Defend by explaining deliverables and answering questions. There should be proof that you did yourself.
Deliverables: report in pdf
Google form:
https://docs.google.com/forms/d/e/1FAIpQLSe0GyNdOYlvM1tX_I_CtlPod5jBf-ACLGdHYZq1gVZbUeBzIg/viewform?usp=sf_link

Exercise 1: Kotlin Syntax Basics

1. **Variables and Data Types:**
   - Create variables of different data types: `Int`, `Double`, `String`, `Boolean`.
   - Print the variables using `println`.



Conditional Statements:

- Create a simple program that checks if a number is positive, negative, or zero.

```kotlin
fun isPositive(number: Int): Any {
    if (number == 0) return "Number is zero"
    return number > 0
}

fun main() {
    val number = readln().toInt()

    val result = isPositive(number)

    if (result == true) {
        println("Number is positive")
    }
    else if (result == false) {
        println("Number is negative")
    }
    else println(result)
}
```

Run ConditionalStatementsKt

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Pro
-1
Number is negative

Process finished with exit code 0
```

Loops:

- Write a program that prints numbers from 1 to 10 using `for` and `while` loops

Main.java Main.kt ConditionalStateme

```kotlin
fun main(){
    var j = 0
    println("For loop:")
    for (i in 1 ≤ .. ≤ 10) {
        print("$i ")
    }
    println()
    println("While loop:")
    while (j < 10){
        j++
        print("$j ")
    }
}
```

Run LoopsKt

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.e
For loop:
1 2 3 4 5 6 7 8 9 10
While loop:
1 2 3 4 5 6 7 8 9 10
Process finished with exit code 0
```

Collections:

- Create a list of numbers, iterate through the list, and print the sum of all numbers.

ConditionalStatements.kt × Loops.kt Collections.kt × VarAndDataT

```kotlin
fun main() {
    val numbers: Array<Int> = arrayOf(1,2,3,4,5)
    var sum = 0
    for (n in numbers) {
        sum += n
        println(n)
    }

    println("Sum of numbers is: $sum")
}
```

Run CollectionsKt ×

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Program F
1
2
3
4
5
Sum of numbers is: 15
```

Exercise 2: Kotlin OOP (Object-Oriented Programming)

1. **Create a `Person` class:**
    - Define properties for `name`, `age`, and `email`.
    - Create a method to display the person's details.

```kotlin
class Person (val name: String, var age: Int, var email: String) {
    override fun toString(): String {
        return "Person(name='$name', age=$age, email='$email')"
    }
}
```

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Program
Person(name='Asset', age=23, email='asset@gmail.com')

Process finished with exit code 0
```

Inheritance:

- Create a class `Employee` that inherits from the `Person` class.
- Add a property for `salary`.
- Override the `displayInfo` method to include the salary.

```kotlin
class Employee(name: String, age: Int, email: String, val salary: Int) : Person(name, age, email) {
    override fun toString(): String {
        return "${super.toString()}, salary= $salary"
    }
}
```

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2
Person name='Asset', age=23, email='asset@gmail.com'
Person name='Abzal', age=23, email='abzal@gmail.com', salary= 1000

Process finished with exit code 0
```

Encapsulation:

- Create a `BankAccount` class with a private property `balance`.

- Provide methods to `deposit` and `withdraw` money, ensuring the balance never goes negative.

Person.kt Employee.kt BankAccount.kt Main.kt

```kotlin
class BankAccount(private var balance: Int) {
    fun deposit(amount: Int) {
        println("Depositing $amount")
        balance += amount
    }
    fun withdraw(amount: Int) {
        println("Withdrawing $amount")
        if (balance >= amount) {
            balance -= amount
        }
        else println("Not enough balance to withdraw, current balance is $balance")
    }
    fun getBalance(): Int {
        return balance
    }
}
```

Run MainKt

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Inte
Withdrawing 80
Balance is: 20
Withdrawing 100
Not enough balance to withdraw, current balance is 20
Balance is: 20
Depositing 100
Balance is: 120

Process finished with exit code 0
```

Exercise 3: Kotlin Functions

1. **Basic Function:**
    - Write a function that takes two integers as arguments and returns their sum

Main.java BasicFunctions.kt

```kotlin
fun main() {
    val a = readln().toInt()
    val b = readln().toInt()
    println("Sum of a and b: ${sumTwo(a,b)}
}

fun sumTwo(a: Int, b: Int): Int {
    return a + b
}
```

Run BasicFunctionsKt

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe
1
3
Sum of a and b: 4

Process finished with exit code 0
```

Lambda Functions:

- Create a lambda function that multiplies two numbers and returns the result

Main.java BasicFunctions.kt LambdaFunctions.kt

```kotlin
fun main() {
    val multiTwo: (Int, Int) -> Int = { a, b -> a * b }

    println(multiTwo(4, 4))
}
```

Run LambdaFunctionsKt

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:
16

Process finished with exit code 0
```

Higher-Order Functions:

- Write a function that takes a lambda function as a parameter and applies it to two integers.

Main.java | BasicFunctions.kt | LambdaFunctions.kt | HIgherOrderFunctions.kt

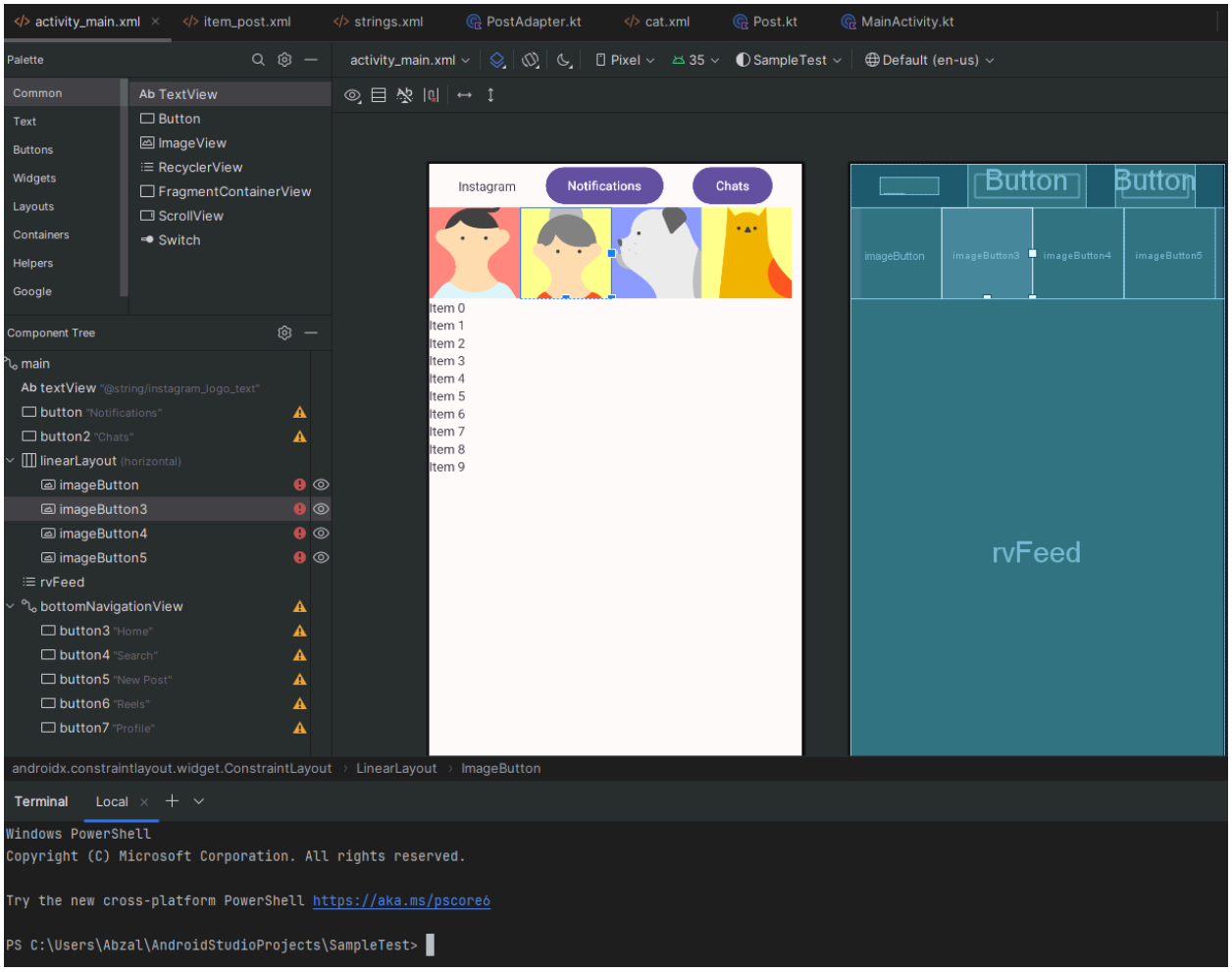
```kotlin
fun main() {
    val compare: (Int, Int) -> Boolean = { a, b -> a > b }

    interpretResult(compare(100, 16))
}

fun interpretResult(compare: Boolean) {
    if (compare) {
        println("First number is greater than second number")
    }
    else println("Second number is greater or equal to First number")
}
```

Run | HIgherOrderFunctionsKt

```
C:\Users\Abzal\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains
First number is greater than second number

Process finished with exit code 0
```

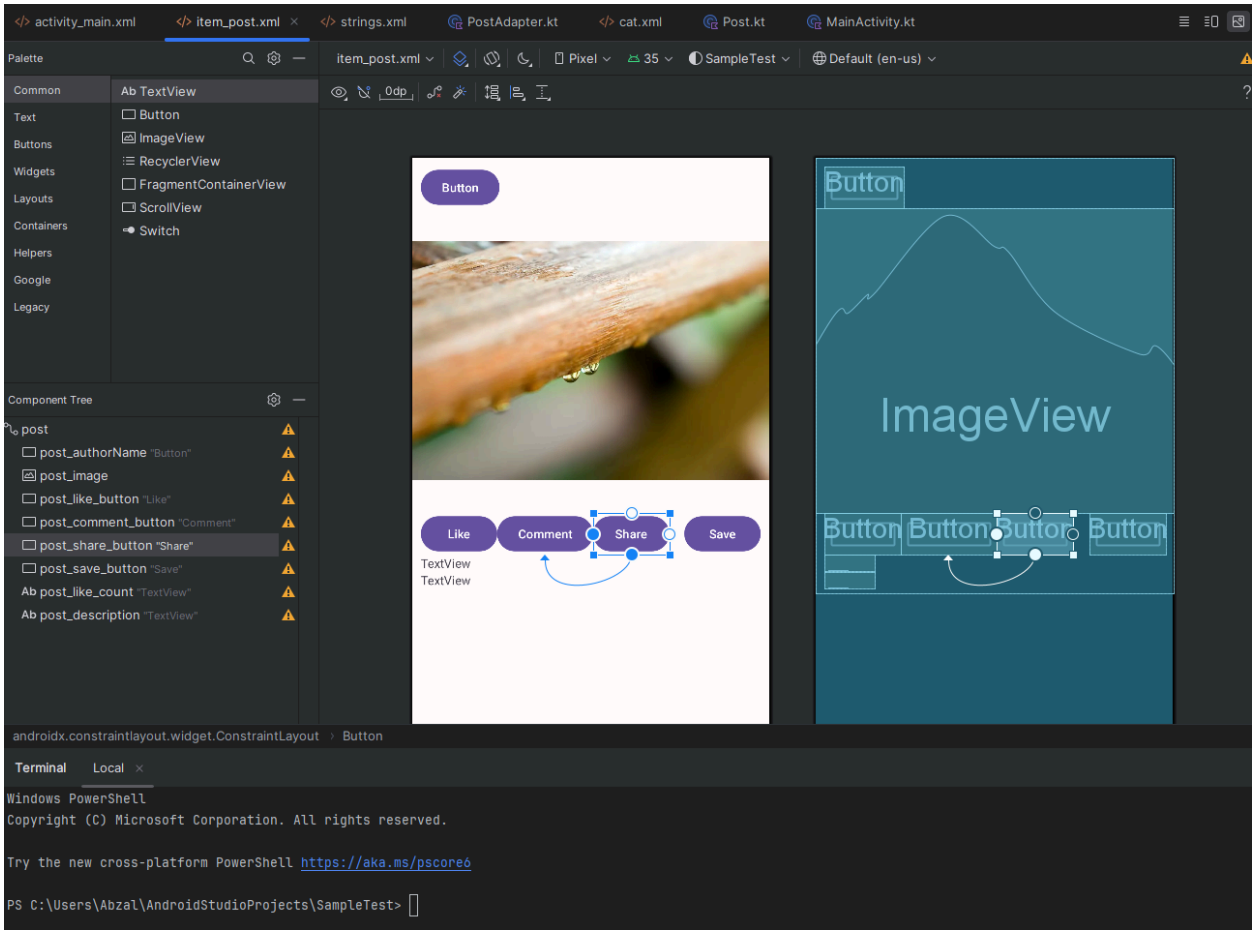
Exercise 4: Android Layout in Kotlin (Instagram-like Layout)

1. **Set Up the Android Project:**
    - Create a new Android project in Android Studio.
    - Ensure you have a Kotlin-based project.
2. **Design the Layout:**
    - Create a new XML layout file (`activity_main.xml`) for a simple Instagram-like user interface.
    - Include elements like `ImageView`, `TextView`, and `RecyclerView` for the feed



Create the RecyclerView Adapter:

- Set up the RecyclerView to display a feed of posts with `ImageView` for the picture and `TextView` for the caption.


activity_main.xml
item_post.xml
strings.xml
PostAdapter.kt
cat.xml
Post.kt
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
FragmentContainerView
ScrollView
Switch
Component Tree
post
post_authorName "Button"
post_image
post_like_button "Like"
post_comment_button "Comment"
post_share_button "Share"
post_save_button "Save"
Ab post_like_count "TextView"
Ab post_description "TextView"
item_post.xml
Pixel
35
SampleTest
Default (en-us)
0dp
Button
Like
Comment
Share
Save
TextView
TextView
Button
ImageView
Button
Button
Button
Button
androidx.constraintlayout.widget.ConstraintLayout
Button
Terminal
Local
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Abzal\AndroidStudioProjects\SampleTest>

activity_main.xml | item_post.xml | strings.xml | PostAdapter.kt | cat.xml | Post.kt | MainActivity.kt

```kotlin
package com.example.sampletest

import ...

class PostAdapter(
    var posts: List<Post>
) : RecyclerView.Adapter<PostAdapter.PostViewHolder>() {

    inner class PostViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val post_author_name: Button = itemView.findViewById(R.id.post_authorName)
        val post_image: ImageView = itemView.findViewById(R.id.post_image)
        val post_like_count: TextView = itemView.findViewById(R.id.post_like_count)
        val post_description: TextView = itemView.findViewById(R.id.post_description)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PostViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.item_post, parent, attachToRoot: false)
        return PostViewHolder(view)
    }

    override fun onBindViewHolder(holder: PostViewHolder, position: Int) {
        holder.apply {
            post_author_name.text = posts[position].author
            post_image.setImageResource(posts[position].image)
            post_like_count.text = posts[position].numOfLikes.toString()
            post_description.text = posts[position].description
        }
    }

    override fun getItemCount(): Int {
        return posts.size
    }
}
```

Terminal | Local

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Abzal\AndroidStudioProjects\SampleTest>
```

MainActivity Setup:

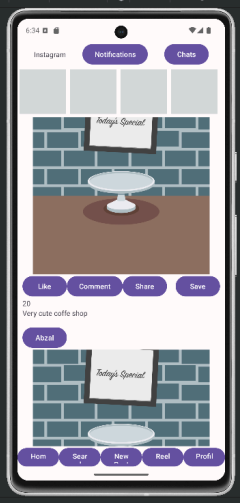
- Initialize the `RecyclerView` in `MainActivity` and populate it with sample data

Sample Test | Version control | Pixel 8a API 35 | app

activity_main.xml | item_post.xml | strings.xml | PostAdapter.kt | cat.xml | Post.kt | MainActivity.kt

```kotlin
package com.example.sampletest

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        val rvFeed: RecyclerView = findViewById(R.id.rvFeed)

        rvFeed.layoutManager = LinearLayoutManager( context: this)

        val postsList = mutableListOf(
            Post( author: "Abzal", R.drawable.cat,  numOfLikes: 20,  description: "Very cute coffe shop"),
            Post( author: "Abzal", R.drawable.cat,  numOfLikes: 20,  description: "Very cute coffe shop"),
            Post( author: "Abzal", R.drawable.cat,  numOfLikes: 20,  description: "Very cute coffe shop"),
            Post( author: "Abzal", R.drawable.cat,  numOfLikes: 20,  description: "Very cute coffe shop"),
            Post( author: "Abzal", R.drawable.cat,  numOfLikes: 20,  description: "Very cute coffe shop"),
            Post( author: "Abzal", R.drawable.cat,  numOfLikes: 20,  description: "Very cute coffe shop")
        )

        val adapter = PostAdapter(postsList)
        rvFeed.adapter = adapter
    }
}
```

Running Devices | Pixel 8a API 35



Terminal | Local

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Abzal\AndroidStudioProjects\SampleTest>
```