Practical I

Go to : https://play.kotlinlang.org/

I. Write a program using Kotlin to implement control structures and loops.

```
fun main() {
    // If-else control structure
    val number = 10
    if (number > 0) {
        println("$number is positive")
    } else if (number < 0) {
        println("$number is negative")
    } else {
        println("$number is zero")
    }

    // When control structure
    val day = 3
    val dayName = when (day) {
        1 -> "Monday"
        2 -> "Tuesday"
        3 -> "Wednesday"
        4 -> "Thursday"
        5 -> "Friday"
        6 -> "Saturday"
        7 -> "Sunday"
        else -> "Invalid day"
    }
    println("Day $day is $dayName")
    // For loop: Print numbers 1 to 5
    println("For loop: Counting from 1 to 5")
    for (i in 1..5) {
        println("Number: $i")
    }
   // While loop: Print numbers 5 to 1
    println("While loop: Counting from 5 to 1")
    var count = 5
    while (count >= 1) {
        println("Number: $count")
        count--
    }
```

```
    // Do-while loop: Execute at least once
    println("Do-while loop example")
    var x = 0
    do {
        println("Value of x: $x")
        x++
    } while (x < 3)

    // For loop with step
    println("For loop with step")
    for (i in 2..10 step 2) {
        println("Even Number: $i")
    }

    // For loop with downTo
    println("For loop with downTo")
    for (i in 5 downTo 1) {
        println("Countdown: $i")
    }

    // Using break
    println("Using break in loop")
    for (i in 1..10) {
        if (i == 6) break  // Stops the loop when i is 6
        println(i)
    }

    // Using continue
    println("Using continue in loop")
    for (i in 1..5) {
        if (i == 3) continue  // Skips number 3
        println(i)
    }
}
```

Output

```
10 is positive
Day 3 is Wednesday
For loop: Counting from 1 to 5
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
While loop: Counting from 5 to 1
Number: 5
Number: 4
Number: 3
Number: 2
Number: 1
Do-while loop example
Value of x: 0
Value of x: 1
Value of x: 2
For loop with step
Even Number: 2
Even Number: 4
Even Number: 6
Even Number: 8
Even Number: 10
For loop with downTo
Countdown: 5
Countdown: 4
Countdown: 3
Countdown: 2
Countdown: 1
Using break in loop
1
2
3
4
5
Using continue in loop
1
2
4
5
```

II. Write a program to implement object-oriented concepts in Kotlin.

```
// Abstract class (Abstraction)
abstract class Vehicle(val name: String, val maxSpeed: Int) {
    abstract fun displayInfo()

    fun generalInfo() {
        println("$name can reach a maximum speed of $maxSpeed km/h.")
    }
}

// Interface
interface Fuel {
    fun fuelType(): String
}

// Base class
open class Car(name: String, maxSpeed: Int, var brand: String) : Vehicle(name,
maxSpeed), Fuel {

    // Property with custom getter & setter (Encapsulation)
    var mileage: Double = 0.0
        get() = field
        set(value) {
            if (value > 0) field = value
            else println("Mileage must be positive!")
        }

    // Primary Constructor (Inheritance)
    init {
        println("$brand $name is created.")
    }

    // Secondary Constructor
    constructor(name: String, maxSpeed: Int, brand: String, mileage: Double) : this(name,
maxSpeed, brand) {
        this.mileage = mileage
    }

    // Overriding abstract function
    override fun displayInfo() {
        println("Car Name: $name, Brand: $brand, Max Speed: $maxSpeed km/h, Mileage:
$mileage km/l")
    }
```

```
    // Implementing interface method
    override fun fuelType(): String {
        return "Petrol/Diesel"
    }
}

// Derived class (Polymorphism & Inheritance)
class ElectricCar(name: String, maxSpeed: Int, brand: String, var batteryLife: Int) :
Car(name, maxSpeed, brand) {

    // Overriding method
    override fun displayInfo() {
        println("Electric Car Name: $name, Brand: $brand, Max Speed: $maxSpeed km/h,
Battery Life: $batteryLife hours")
    }

    // Overriding interface method
    override fun fuelType(): String {
        return "Electric Battery"
    }
}

// Main function to test the OOP implementation
fun main() {
    // Creating an instance of Car
    val myCar = Car("Sedan", 220, "Toyota", 15.5)
    myCar.displayInfo()
    myCar.generalInfo()
    println("Fuel Type: ${myCar.fuelType()}")

    println("\n----------------------\n")

    // Creating an instance of ElectricCar
    val myEV = ElectricCar("Model S", 250, "Tesla", 24)
    myEV.displayInfo()
    myEV.generalInfo()
    println("Fuel Type: ${myEV.fuelType()}")
}
```

Output

Toyota Sedan is created.
Car Name: Sedan, Brand: Toyota, Max Speed: 220 km/h, Mileage: 15.5 km/l
Sedan can reach a maximum speed of 220 km/h.
Fuel Type: Petrol/Diesel

Tesla Model S is created.
Electric Car Name: Model S, Brand: Tesla, Max Speed: 250 km/h, Battery Life: 24 hours
Model S can reach a maximum speed of 250 km/h.
Fuel Type: Electric Battery

Practical 2

Create an Android application to design screens using different layouts and UI including Edittext, Textview.

Step I : Create a New Android Project

- Open **Android Studio**.
- Click **"New Project"** → Select **"Empty Views Activity"**.
- Name the project, for example, **"UIPractical"**.
- Choose **Kotlin** as the language.
- Click **Finish** to create the project.

Step 2: open `activity_main.xml` (User Interface)

- Open **`res/layout/activity_main.xml`**.
- Replace with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/edtName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="108dp"
        android:layout_marginTop="120dp"
        android:layout_marginEnd="93dp"
        android:layout_marginBottom="42dp"
        android:ems="10"
        android:hint="Name"
        android:inputType="text"
        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/butShow"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/butShow"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginStart="156dp"
        android:layout_marginTop="42dp"
        android:layout_marginEnd="165dp"
        android:text="Show"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edtName" />

    <TextView
        android:id="@+id/txtMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="176dp"
        android:layout_marginTop="435dp"
        android:layout_marginEnd="178dp"
        android:layout_marginBottom="129dp"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/butShow" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 3: Modify `MainActivity.kt`

- Open **`MainActivity.kt`**.
- Replace the content with:

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView

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

        val butShow = findViewById<Button>(R.id.butShow);

        val txtMsg = findViewById<TextView>(R.id.txtMessage);
        val edtName = findViewById<EditText>(R.id.edtName);
        butShow.setOnClickListener(){
            val value= edtName.text;
            txtMsg.setText(""+value);
        }
    }
```

```
}
```

Practical 3(a)

Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image.

Step I : Navigate to the app > res > layout > activity_main.xml and add the below code to that file. Below is the code for the activity_main.xml file.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<!--on below line we are adding view pager -->
<androidx.viewpager.widget.ViewPager
        android:id="@+id/idViewPager"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerInParent="true"
        android:layout_gravity="center"
        android:contentDescription="View Pager "
        android:layout_margin="10dp" />
</RelativeLayout>
```

Step II Creating a layout file for ImageView in View Pager
Navigate to the app > res > layout > Right-click on it > New > Layout Resource file and specify the name as image_slider_item. Add the below code to it.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
<!--on below line we are creating an image view-->
        <ImageView
        android:id="@+id/idIVImage"
```

```
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

Step III Creating a new java class for the adapter of our ViewPager
Navigate to the app > java > your app's package name > Right-click on it > New > Java/Kotlin class and name it as ViewPagerAdapter and add the below code to it.

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.RelativeLayout
import androidx.viewpager.widget.PagerAdapter
import java.util.*
class ViewPagerAdapter(val context: Context, val imageList: List<Int>) : PagerAdapter() {

    override fun getCount(): Int {
        return imageList.size
    }

    override fun isViewFromObject(view: View, `object`: Any): Boolean {
        return view === `object` as RelativeLayout
    }

    override fun instantiateItem(container: ViewGroup, position: Int): Any {

        val mLayoutInflater =
            context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater

        val itemView: View = mLayoutInflater.inflate(R.layout.image_slider_item, container,
false)
        val imageView: ImageView = itemView.findViewById<View>(R.id.idIVImage) as ImageView
        imageView.setImageResource(imageList.get(position))
        Objects.requireNonNull(container).addView(itemView)
        return itemView
    }

    override fun destroyItem(container: ViewGroup, position: Int, `object`: Any) {
        container.removeView(`object` as RelativeLayout)
    }
}
```

Step IV Adding images to the drawable folder Select the images which you want to add copy them Navigate to app > res > drawable and right-click on it. Simply paste it and add all the images to the drawable folder.

Step V Working with the MainActivity.kt file
Go to the MainActivity.kt file and refer to the following code. Below is the code for the MainActivity.kt file.

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.viewpager.widget.ViewPager

class MainActivity : AppCompatActivity() {
    lateinit var viewPager: ViewPager
    lateinit var viewPagerAdapter: ViewPagerAdapter
    lateinit var imageList: List<Int>
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        viewPager = findViewById(R.id.idViewPager)
        imageList = ArrayList<Int>()
        imageList = imageList + R.drawable.img
        imageList = imageList + R.drawable.img_1
        imageList = imageList + R.drawable.img_2
        imageList = imageList + R.drawable.img_3
        imageList = imageList + R.drawable.img_4
        viewPagerAdapter = ViewPagerAdapter(this@MainActivity, imageList)
        viewPager.adapter = viewPagerAdapter
    }
}
```

Program 3(ii)-
Create an application to use Gridview for shopping cart application

Step I Add the following code to res/layout/activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <GridView
        android:id="@+id/gridView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:numColumns="2" />
</RelativeLayout>
```

Step II Add the following code to src/MainActivity.kt

```
package com.example.gridview

import android.os.Bundle
import android.widget.AdapterView.OnItemClickListener
import android.widget.GridView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
    lateinit var gridView: GridView
    private var playerNames = arrayOf("Cristiano Ronaldo",
        "Joao Felix",
        "Bernado Silva",
        "Andre Silve",
        "Bruno Fernandez",
        "William Carvalho",
        "Nelson Semedo",
        "Pepe",
        "Rui Patricio")
    private var playerImages = intArrayOf(R.drawable.ronaldo,
        R.drawable.felix,
        R.drawable.bernado,
        R.drawable.andre,
        R.drawable.bruno,
        R.drawable.carvalho,
        R.drawable.semedo,
        R.drawable.pepe,
        R.drawable.patricio)

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        title = "KotlinApp"
        gridView = findViewById(R.id.gridView)
        val mainAdapter = MainAdapter(this@MainActivity, playerNames, playerImages)
        gridView.adapter = mainAdapter
        gridView.onItemClickListener = OnItemClickListener { _, _, position, _ ->
            Toast.makeText(applicationContext, "You CLicked " + playerNames[+position],
                Toast.LENGTH_SHORT).show()
        }
    }
```

```
}
```

Step III − Create a Kotlin class (MyAdapter.kt) and add the following code

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.BaseAdapter
import android.widget.ImageView
import android.widget.TextView
internal class MainAdapter(
   private val context: Context,
   private val numbersInWords: Array<String>,
   private val numberImage: IntArray
) :
   BaseAdapter() {
   private var layoutInflater: LayoutInflater? = null
   private lateinit var imageView: ImageView
   private lateinit var textView: TextView
   override fun getCount(): Int {
       return numbersInWords.size
   }
   override fun getItem(position: Int): Any? {
       return null
   }
   override fun getItemId(position: Int): Long {
       return 0
   }
   override fun getView(
       position: Int,
       convertView: View?,
       parent: ViewGroup
   ): View? {
       var convertView = convertView
       if (layoutInflater == null) {
           layoutInflater =
               context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater
       }
       if (convertView == null) {
           convertView = layoutInflater!!.inflate(R.layout.row_item, null)
       }
       imageView = convertView!!.findViewById(R.id.imageView)
       textView = convertView.findViewById(R.id.textView)
       imageView.setImageResource(numberImage[position])
       textView.text = numbersInWords[position]
       return convertView
   }
}
```

Step IV − Create a Layout Resource file (row_item.xml) and add the following code −

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:orientation="vertical"
```

```
    android:gravity="center"
    android:padding="8dp">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="100dp"
        android:layout_height="100dp" />
    <TextView
        android:textAlignment="center"
        android:gravity="center"
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Numbers"
        android:layout_marginBottom="10dp"
        android:textColor="@android:color/background_dark"
        android:textSize="24sp"
        android:textStyle="bold" />
</LinearLayout>
```

Practical 4

Create an Android application to demonstrate implicit and explicit intents

Step I : Create a New Android Project

- Open **Android Studio**.
- Click **"New Project"** → Select **"Empty Views Activity"**.
- Name the project, for example, **"IntentsPractical"**.
- Choose **Kotlin** as the language.
- Click **Finish** to create the project.

Step 2: open `activity_main.xml` (User Interface)

- Open **`res/layout/activity_main.xml`**.
- Replace with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context=".MainActivity" >
   <EditText
       android:id="@+id/editURL"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_marginLeft="10dp"
       android:layout_marginTop="100dp"
       android:layout_marginRight="20dp"
       android:ems="10"
       android:inputType="text" />
   <Button
       android:id="@+id/butImplicit"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_marginTop="20dp"
       android:text="Implicit Intent" />
   <Button
       android:id="@+id/butExplicit"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_marginTop="20dp"
       android:text="Explicit Intent" />
</LinearLayout>
```

Step III : Create a another activity

Right Click app -> New -> Activity -> EmptyViewsActivity -> name as “SecondActivity.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="100dp"
        android:layout_marginEnd="50dp"
        android:layout_marginBottom="383dp"
        android:text="Hello World"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 5: Modify `MainActivity.kt`

- Open **`MainActivity.kt`**.
- Replace the content with:

```
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
```

```
        val editUrl = findViewById<EditText>(R.id.editURL)
        val butImplicit = findViewById<Button>(R.id.butImplicit)
        val butExplicit = findViewById<Button>(R.id.butExplicit)

        butImplicit.setOnClickListener()
        {
            val url = editUrl.text.toString()
            val urlIntent = Intent(Intent.ACTION_VIEW, Uri.parse(url))
            startActivity(urlIntent)
        }

        butExplicit.setOnClickListener(){
            val secondIntent = Intent(this,SecondActivity::class.java)
            startActivity(secondIntent)
        }

    }
}
```

Practical 5

Create an Android application to demonstrate the use of Broadcast listeners.

Step I : Create a New Android Project

- Open **Android Studio**.
- Click **"New Project"** → Select **"Empty Views Activity"**.
- Name the project, for example, **"BroadcastPractical"**.
- Choose **Kotlin** as the language.
- Click **Finish** to create the project.

Step II : Create File Named AirPlaneModeChangeReceiver.kt

Select kotlin+java from app -> Right click on package com.example.**BroadcastPractical** folder -> New -> Kotlin Class/File -> name it as “AirPlaneModeChangeReceiver.kt”

Step III : Add The Following Code

```
import android.content.BroadcastReceiver
import android.content.ContentValues.TAG
import android.content.Context
import android.content.Intent
import android.util.Log
import android.widget.Toast
class AirplaneModeChangeReceiver : BroadcastReceiver() {
   override fun onReceive(context: Context?, intent: Intent?) {
       val isAirplaneModeEnabled = intent?.getBooleanExtra("state", false) ?: return

       if (isAirplaneModeEnabled) {
           Toast.makeText(context, "Airplane Mode Enabled", Toast.LENGTH_LONG).show()
       } else {
           Toast.makeText(context, "Airplane Mode Disabled", Toast.LENGTH_LONG).show()
       }
   }
}
```

Step IV : Open MainActivity.kt file

```
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
   lateinit var receiver: AirplaneModeChangeReceiver
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       enableEdgeToEdge()
       setContentView(R.layout.activity_main)
       receiver = AirplaneModeChangeReceiver()
       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
       }
      IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED).also {
           registerReceiver(receiver, it)
       }
   }
   override fun onStop() {
       super.onStop()
       unregisterReceiver(receiver)
   }
}
```

Practical 6

Create an Android application to demonstrate XML based animation

Step I : Create a New Android Project

- Open **Android Studio**.
- Click **"New Project"** → Select **"Empty Views Activity"**.
- Name the project, for example, **"AnimationPractical"**.
- Choose **Kotlin** as the language.
- Click **Finish** to create the project.

Step II : Create animation.xml file

- Right click on res -> New -> Android Resource File -> Name as “animation.xml” and Resource type as “Animation”
- Replace with following Content

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:fillAfter="true"
   android:interpolator="@android:anim/bounce_interpolator">

   <scale
       android:pivotX="50%"
       android:pivotY="50%"
       android:fromXScale="0.5"
       android:toXScale="1.0"
       android:fromYScale="0.5"
       android:toYScale="1.0"
       android:duration="500" />
</set>
```

Step III : Modify activity_main.xml file

- Replace with following Content

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context=".MainActivity" >
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="250dp"
        android:text="Start Animation" />
</LinearLayout>
```

Step IV : Modify `MainActivity.kt file`

- Replace with following content

```
import android.os.Bundle
import android.view.animation.Animation
import android.view.animation.AnimationUtils
import android.widget.Button
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        val animation: Animation =
AnimationUtils.loadAnimation(this,R.anim.animation)
        val button : Button = findViewById<Button>(R.id.button)

        button.setOnClickListener(){
            button.startAnimation(animation)
        }
    }
}
```

Practical 7

Create an Android application to demonstrate the different types of menus.
a. Pop-up Menu
b. Context Menu
c. Option Menu

Step I : Create a New Android Project

- Open **Android Studio**.
- Click **"New Project"** → Select **"Empty Views Activity"**.
- Name the project, for example, **"MenusPractical"**.
- Choose **Kotlin** as the language.
- Click **Finish** to create the project.

Step 2: Modify `activity_main.xml`

1. Open **`res/layout/activity_main.xml`**.
2. Replace the content with:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center"
   android:padding="16dp">
   <!-- Button for Pop-up Menu -->
   <Button
       android:id="@+id/btnPopupMenu"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Show Pop-up Menu"/>
   <!-- TextView for Context Menu -->
   <TextView
       android:id="@+id/tvContextMenu"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Long Press Me (Context Menu)"
       android:textSize="18sp"
       android:padding="16dp"
       android:background="@android:color/darker_gray"
       android:layout_marginTop="20dp"/>

</LinearLayout>
```

Step 3: Create Menu XML Files

1. Create `popup_menu.xml` (Pop-up Menu)

1. Right Click on res folder -> New -> Android Resource File -> Name as "**`popup_menu.xml`**" and Resource type as "Menu"
2. Add the following:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/item1" android:title="Item 1"/>
    <item android:id="@+id/item2" android:title="Item 2"/>
    <item android:id="@+id/item3" android:title="Item 3"/>
</menu>
```

2. Create `context_menu.xml` (Pop-up Menu)

3. Right Click on res folder -> New -> Android Resource File -> Name as "**`context_menu.xml`**" and Resource type as "Menu"
4. Add the following:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/context_edit" android:title="Edit"/>
    <item android:id="@+id/context_delete" android:title="Delete"/>
</menu>
```

3. Create `options_menu.xml` (Pop-up Menu)

5. Right Click on res folder -> New -> Android Resource File -> Name as "**`options_menu.xml`**" and Resource type as "Menu"
6. Add the following:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/settings" android:title="Settings"/>
    <item android:id="@+id/about" android:title="About"/>
</menu>
```

Step 4: Modify `MainActivity.kt`

1. Open **`MainActivity.kt`**.
2. Replace the content with:

```
import android.os.Bundle
import android.view.ContextMenu
import android.view.Menu
import android.view.View
import android.widget.Button
import android.widget.PopupMenu
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val btnPopupMenu: Button = findViewById(R.id.btnPopupMenu)
        val tvContextMenu: TextView = findViewById(R.id.tvContextMenu)
        registerForContextMenu(tvContextMenu)
        btnPopupMenu.setOnClickListener {
            showPopupMenu(it)
        }
    }
    // Create Options Menu
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.options_menu, menu)
        return true
    }
    // Show Pop-up Menu
    private fun showPopupMenu(view: View) {
        val popup = PopupMenu(this, view)
        popup.menuInflater.inflate(R.menu.popup_menu, popup.menu)
        popup.show()
    }
    // Create Context Menu
    override fun onCreateContextMenu(menu: ContextMenu, v: View, menuInfo:
ContextMenu.ContextMenuInfo?) {
        super.onCreateContextMenu(menu, v, menuInfo)
        menuInflater.inflate(R.menu.context_menu, menu)
    }
}
```

Step 4: Modify `AndroidManifest.xml`

1. Open app -> manifest -> **`AndroidManifest.xml`**
2. Add or replace the following line :

android:theme="@style/Theme.AppCompat.Light.DarkActionBar"

Practical 8

Create a suitable Android application to store and retrieve data in the SQLite database.

Step I : Create a New Android Project

- Open **Android Studio**.
- Click **"New Project"** → Select **"Empty Views Activity"**.
- Name the project, for example, **"SQLitePractical"**.
- Choose **Kotlin** as the language.
- Click **Finish** to create the project.

Step 2: Modify `AndroidManifest.xml`

1. **Add permissions** (not required for local SQLite, but useful for storage-related operations):

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Step 3: open `activity_main.xml` (User Interface)

- Open **`res/layout/activity_main.xml`**.
- Replace with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:padding="16dp">

   <!-- EditText to enter name -->
   <EditText
       android:id="@+id/etName"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:hint="Enter Name"/>

   <!-- Button to insert data -->
```

```
    <Button
        android:id="@+id/btnInsert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Insert Data"/>

    <!-- Button to fetch data -->
    <Button
        android:id="@+id/btnFetch"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Fetch Data"
        android:layout_marginTop="8dp"/>

    <!-- TextView to display data -->
    <TextView
        android:id="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:padding="8dp"/>
</LinearLayout>
```

Step 4: Create `DatabaseHelper.kt` (SQLite Database Helper Class)

1. **Right-click on `app/java/com.example.sqlitepractical/` → New → Kotlin Class** → Name it **`DatabaseHelper`**.
2. **Add the following code**:

```
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DatabaseHelper(context: Context) : SQLiteOpenHelper(context,
DATABASE_NAME, null, 1) {

    companion object {
        private const val DATABASE_NAME = "UserDB"
        private const val TABLE_NAME = "users"
        private const val COLUMN_ID = "id"
        private const val COLUMN_NAME = "name"
    }

    override fun onCreate(db: SQLiteDatabase) {
```

```
        val createTable = "CREATE TABLE $TABLE_NAME ($COLUMN_ID INTEGER PRIMARY
KEY AUTOINCREMENT, $COLUMN_NAME TEXT)"
        db.execSQL(createTable)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int)
{
        db.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }

    // Insert Data
    fun insertData(name: String): Boolean {
        val db = writableDatabase
        val contentValues = ContentValues()
        contentValues.put(COLUMN_NAME, name)
        val result = db.insert(TABLE_NAME, null, contentValues)
        return result != -1L
    }

    // Fetch Data
    fun getAllData(): Cursor {
        val db = readableDatabase
        return db.rawQuery("SELECT * FROM $TABLE_NAME", null)
    }
}
```

Step 5: Modify `MainActivity.kt`

- Open **`MainActivity.kt`**.
- Replace the content with:

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.database.Cursor
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var etName: EditText
    private lateinit var btnInsert: Button
    private lateinit var btnFetch: Button
    private lateinit var tvResult: TextView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
```

```
        dbHelper = DatabaseHelper(this)
        etName = findViewById(R.id.etName)
        btnInsert = findViewById(R.id.btnInsert)
        btnFetch = findViewById(R.id.btnFetch)
        tvResult = findViewById(R.id.tvResult)
        // Insert Data
        btnInsert.setOnClickListener {
            val name = etName.text.toString()
            if (name.isNotEmpty()) {
                val isInserted = dbHelper.insertData(name)
                if (isInserted) {
                    Toast.makeText(this, "Data Inserted",
Toast.LENGTH_SHORT).show()
                    etName.text.clear()
                } else {
                    Toast.makeText(this, "Insertion Failed",
Toast.LENGTH_SHORT).show()
                }
            } else {
                Toast.makeText(this, "Enter a name!", Toast.LENGTH_SHORT).show()
            }
        }
        // Fetch Data
        btnFetch.setOnClickListener {
            val cursor: Cursor = dbHelper.getAllData()
            val stringBuilder = StringBuilder()
            if (cursor.moveToFirst()) {
                do {
                    val id = cursor.getInt(0)
                    val name = cursor.getString(1)
                    stringBuilder.append("ID: $id, Name: $name\n")
                } while (cursor.moveToNext())
            } else {
                stringBuilder.append("No data found")
            }
            tvResult.text = stringBuilder.toString()
        }
    }
}
```