|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Advanced Mobile Programming  Practical #8 | | |
|  |  |  |  |
| **Name** | **Aniket Gupta** | **Roll Number** | **21302B0053** |
| **Subject/Course:** | **Advanced Mobile Programming** | | |
| **Topic** | **Notification and Receiver** | | |
|  | | | |
| **Notification** | | | |
| 8.a) Design an Android mobile application to demonstrate the working of notifications. [Hint: create and display the notification with help of button] | | | |
| **MainActivity.kt**  package com.example.practical8  // Import necessary Android packages  import android.app.NotificationChannel  import android.app.NotificationManager  import android.content.Context  import android.os.Build  import android.support.v7.app.AppCompatActivity  import android.os.Bundle  import android.support.v4.app.NotificationCompat  import android.support.v4.app.NotificationManagerCompat  import android.widget.Button  // MainActivity class extending AppCompatActivity  class MainActivity : AppCompatActivity() {  // Define constants for notification channel ID and notification ID  private val channelId = "sample\_channel"  private val notificationId = 101   // Override onCreate method  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)   // Find the notifyButton from the layout  val button = findViewById<Button>(R.id.notifyButton)   // Set click listener for the button  button.setOnClickListener {  // Create notification channel before sending notification  createNotificationChannel()  // Send notification  sendNotification()  }  }   // Function to create notification channel  private fun createNotificationChannel() {  // Check if the Android version is Oreo or higher  if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {  val name = "Sample Channel"  val descriptionText = "This is a sample notification channel."  val importance = NotificationManager.IMPORTANCE\_DEFAULT  // Create notification channel  val channel = NotificationChannel(channelId, name, importance).apply {  description = descriptionText  }  // Get NotificationManager system service  val notificationManager: NotificationManager =  getSystemService(Context.NOTIFICATION\_SERVICE) as NotificationManager  // Create the notification channel  notificationManager.createNotificationChannel(channel)  }  }   // Function to send notification  private fun sendNotification() {  // Build the notification using NotificationCompat Builder  val builder = NotificationCompat.Builder(this, channelId)  .setSmallIcon(R.drawable.ic\_launcher\_background) // Set notification icon  .setContentTitle("Sample Notification") // Set notification title  .setContentText("This is a sample notification.") // Set notification content text  .setPriority(NotificationCompat.PRIORITY\_DEFAULT) // Set notification priority    // Notify the notification using NotificationManagerCompat  with(NotificationManagerCompat.from(this)) {  notify(notificationId, builder.build())  }  }  }  **activity\_main.xml**  <?xml version="1.0" encoding="utf-8"?> <RelativeLayout  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">   <Button  android:id="@+id/notifyButton"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Show Notification"  android:layout\_centerInParent="true"  tools:layout\_editor\_absoluteX="154dp"  tools:layout\_editor\_absoluteY="417dp" /> </RelativeLayout>  **OUTPUT:** | | | |
|  | | | |

|  |
| --- |
| **Broadcast Receiver** |
| 8.b) Design an Android mobile application to show the working of broadcast receiver. |
| **AndroidManifest.xml**  *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **package="com.example.amp\_p82\_1"**>   <**uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"** />   <**application**  **android:allowBackup="true"**  **android:dataExtractionRules="@xml/data\_extraction\_rules"**  **android:fullBackupContent="@xml/backup\_rules"**  **android:icon="@mipmap/ic\_launcher"**  **android:label="@string/app\_name"**  **android:roundIcon="@mipmap/ic\_launcher\_round"**  **android:supportsRtl="true"**  **android:theme="@style/Theme.AMP\_P82\_1"**  **tools:targetApi="31"**>   *<!-- MainActivity configuration -->* <**activity**  **android:name=".MainActivity"**  **android:exported="true"**>  <**intent-filter**>  <**action android:name="android.intent.action.MAIN"** />  <**category android:name="android.intent.category.LAUNCHER"** />  </**intent-filter**>  </**activity**>   *<!-- Register the receiver for airplane mode changes -->* <**receiver android:name=".AirplaneModeReceiver"**  **android:exported="true"**>  <**intent-filter**>  *<!-- Broadcast receiver listens for airplane mode changes -->* <**action android:name="android.intent.action.AIRPLANE\_MODE"** />  </**intent-filter**>  </**receiver**>  </**application**>  </**manifest**>  **activity\_main.xml**  *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.constraint.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"**  **android:background="@color/white"**  **tools:context=".MainActivity"**>   <**TextView**  **android:id="@+id/broadcastStatus"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Broadcast Status"**  **android:textSize="24sp"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent"** />  </**android.support.constraint.ConstraintLayout**>  **MainActivity.kt**  **package** com.example.amp\_p82\_1  **import** android.content.Intent **import** android.content.IntentFilter **import** android.support.v7.app.AppCompatActivity **import** android.os.Bundle  **class** MainActivity : AppCompatActivity() {  **private val airplaneModeReceiver** = AirplaneModeReceiver()  **override fun** onCreate(savedInstanceState: Bundle?) {  **super**.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  registerReceiver(**airplaneModeReceiver**, IntentFilter(Intent.*ACTION\_AIRPLANE\_MODE\_CHANGED*))  }   **override fun** onDestroy() {  **super**.onDestroy()  unregisterReceiver(**airplaneModeReceiver**)  } }  **airplaneModeReceiver.kt**  **package** com.example.amp\_p82\_1  **import** android.content.BroadcastReceiver **import** android.content.Context **import** android.content.Intent **import** android.provider.Settings **import** android.widget.Toast  **class** AirplaneModeReceiver : BroadcastReceiver() {  **override fun** onReceive(context: Context?, intent: Intent?) {  **if** (intent?.*action* == Intent.*ACTION\_AIRPLANE\_MODE\_CHANGED*) {  **val** isAirplaneModeOn = isAirplaneModeOn(context)  **val** message = **if** (isAirplaneModeOn) **"Airplane mode is ON" else "Airplane mode is OFF"** showToast(context, message)  }  }   **private fun** isAirplaneModeOn(context: Context?): Boolean {  **return** Settings.Global.getInt(context?.*contentResolver*, Settings.Global.*AIRPLANE\_MODE\_ON*, 0) != 0  }   **private fun** showToast(context: Context?, message: String) {  Toast.makeText(context, message, Toast.*LENGTH\_SHORT*).show()  } } |
| **OUTPUT:** |