

Create an android application to generate two notifications, one notification will be non-clickable and another is clickable (it will reload the current application)

NotificationExample filename

In your AndroidManifest.xml, add:

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

MainActivity.kt

```
package com.example.notificationexample

import android.app.*
import android.content.Context
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.widget.Button
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.NotificationCompat
import androidx.core.content.ContextCompat
```

```
class MainActivity : AppCompatActivity() {

    private val CHANNEL_ID = "default_channel"

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

        createNotificationChannel()

        // Ask for permission (Android 13+)
        askNotificationPermission()

        val buttonNotify = findViewById<Button>(R.id.btnNotify)
        buttonNotify.setOnClickListener{
            sendNonClickableNotification()
            sendClickableNotification()
        }
    }

    private fun createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val name = "Default Channel"
            val descriptionText = "This is the default notification channel"
            val importance = NotificationManager.IMPORTANCE_DEFAULT
            val channel = NotificationChannel(CHANNEL_ID, name, importance).apply {
                description = descriptionText
            }
        }
    }
}
```

```
    val notificationManager: NotificationManager =
        getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
    notificationManager.createNotificationChannel(channel)

}

}

private fun sendNonClickableNotification() {
    val notification = NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.ic_launcher_foreground)
        .setContentTitle("Non-Clickable Notification")
        .setContentText("This notification does nothing when clicked.")
        .setAutoCancel(true)
        .build()

    val notificationManager = getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
    notificationManager.notify(1, notification)
}

private fun sendClickableNotification() {
    val intent = Intent(this, MainActivity::class.java).apply {
        flags = Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK
    }

    val pendingIntent: PendingIntent = PendingIntent.getActivity(
        this, 0, intent,
        PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
    )
}
```

```
val notification = NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.ic_launcher_foreground)
    .setContentTitle("Clickable Notification")
    .setContentText("Clicking will reload the app.")
    .setContentIntent(pendingIntent)
    .setAutoCancel(true)
    .build()

    val notificationManager = getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
    notificationManager.notify(2, notification)
}

private fun askNotificationPermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
            val requestPermissionLauncher = registerForActivityResult(
                ActivityResultContracts.RequestPermission()
            ) { isGranted: Boolean ->
                if (!isGranted) {
                    // Handle denial
                }
            }
            requestPermissionLauncher.launch(android.Manifest.permission.POST_NOTIFICATIONS)
        }
    }
}
```

```
    }  
}  
}  
}
```

activity_main.xml

```
<?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:gravity="center"  
    android:orientation="vertical"  
    android:padding="24dp">  
  
<Button  
    android:id="@+id	btnNotify"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Show Notifications" />  
</LinearLayout>
```

Create an android application to display Alert Dialog on pressing the Back button. for android studio

BackButtonAlertApp

MainActivity.kt

```
package com.example.backbuttonalertapp

import android.app.AlertDialog
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

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

    override fun onBackPressed() {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Exit App")
        builder.setMessage("Are you sure you want to exit?")
        builder.setPositiveButton("Yes") { dialog, _ ->
            dialog.dismiss()
        }
    }
}
```

```
super.onBackPressed()

}

builder.setNegativeButton("No") { dialog, _ ->
    dialog.dismiss()
}

builder.setCancelable(false)
builder.show()

}

}
```

activity_main.xml

```
<?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:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Press Back to see the Alert Dialog"  
        android:textSize="18sp"/>  
  
</LinearLayout>
```

Create the standard calculator application in android studio

CalculatorApp

res/layout/activity_main.xml

```
<?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"  
    android:gravity="center">
```

<TextView

```
    android:id="@+id/tvInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="32sp"  
    android:gravity="end"  
    android:text=""/>
```

```
<GridLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:columnCount="4"  
    android:rowCount="5"  
    android:layout_marginTop="16dp">  
  
    <!-- Number buttons -->  
    <Button android:id="@+id	btn7" android:text="7"/>  
    <Button android:id="@+id	btn8" android:text="8"/>  
    <Button android:id="@+id	btn9" android:text="9"/>  
    <Button android:id="@+id	btnDivide" android:text="/" />  
  
    <Button android:id="@+id	btn4" android:text="4"/>  
    <Button android:id="@+id	btn5" android:text="5"/>  
    <Button android:id="@+id	btn6" android:text="6"/>  
    <Button android:id="@+id	btnMultiply" android:text="*"/>  
  
    <Button android:id="@+id	btn1" android:text="1"/>  
    <Button android:id="@+id	btn2" android:text="2"/>  
    <Button android:id="@+id	btn3" android:text="3"/>  
    <Button android:id="@+id	btnSubtract" android:text="-"/>  
  
    <Button android:id="@+id	btnClear" android:text="C"/>  
    <Button android:id="@+id	btn0" android:text="0"/>  
    <Button android:id="@+id	btnEqual" android:text="="/>  
    <Button android:id="@+id	btnAdd" android:text="+"/>
```

```
</GridLayout>  
</LinearLayout>
```

MainActivity.kt

```
package com.example.calculatorapp  
  
import android.os.Bundle  
import android.widget.Button  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
import java.util.*  
  
class MainActivity : AppCompatActivity() {  
    private lateinit var tvInput: TextView  
    private var currentExpression = ""  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        tvInput = findViewById(R.id.tvInput)  
  
        val buttons = listOf(  
            R.id.btn0, R.id.btn1, R.id.btn2, R.id.btn3, R.id.btn4,  
            R.id.btn5, R.id.btn6, R.id.btn7, R.id.btn8, R.id.btn9,  
            R.id.btnAdd, R.id.btnSubtract, R.id.btnMultiply, R.id.btnDivide
```

```
)  
  
for (id in buttons) {  
    findViewById<Button>(id).setOnClickListener {  
        val text = (it as Button).text.toString()  
        currentExpression += text  
        tvInput.text = currentExpression  
    }  
}  
  
findViewById<Button>(R.id.btnClear).setOnClickListener {  
    currentExpression = ""  
    tvInput.text = ""  
}  
  
findViewById<Button>(R.id.btnEqual).setOnClickListener {  
    try {  
        val result = evaluate(currentExpression)  
        tvInput.text = result.toString()  
        currentExpression = result.toString()  
    } catch (e: Exception) {  
        tvInput.text = "Error"  
        currentExpression = ""  
    }  
}  
  
private fun evaluate(expression: String): Double {
```

```
val tokens = Stack<Double>()
val operations = Stack<Char>()
var i = 0

while (i < expression.length) {

    val c = expression[i]

    if (c.isDigit() || c == '.') {

        var num = ""

        while (i < expression.length && (expression[i].isDigit() || expression[i] == '.')) {

            num += expression[i]

            i++
        }

        tokens.push(num.toDouble())
        i--
    } else if (c in "+-*/") {

        while (operations.isNotEmpty() && precedence(c) <=
precedence(operations.peek())) {

            val b = tokens.pop()

            val a = tokens.pop()

            val op = operations.pop()

            tokens.push(applyOp(a, b, op))
        }

        operations.push(c)
    }
    i++
}

while (operations.isNotEmpty()) {
```

```
    val b = tokens.pop()
    val a = tokens.pop()
    val op = operations.pop()
    tokens.push(applyOp(a, b, op))
}

return tokens.pop()
}

private fun precedence(op: Char): Int {
    return when (op) {
        '+', '-' -> 1
        '*', '/' -> 2
        else -> 0
    }
}

private fun applyOp(a: Double, b: Double, op: Char): Double {
    return when (op) {
        '+' -> a + b
        '-' -> a - b
        '*' -> a * b
        '/' -> a / b
        else -> 0.0
    }
}
}
```

Create an android application for the following menu items ,the appropriate toast should appear by clicking on the item : • Settings • Search • Compose Email • FeedBack (make Compose Email item disabled)

MenuApp

- ② Go to: res > menu (If it doesn't exist: Right click res → New → Android Resource Directory → Name: menu, Resource type: menu)
 - ② Right click on menu folder → New → Menu Resource File → Name it: main_menu.xml
-

```
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/menu_settings"
        android:title="Settings" />

    <item
        android:id="@+id/menu_search"
        android:title="Search" />

    <item
        android:id="@+id/menu_email"
        android:title="Compose Email"
        android:enabled="false" />

    <item
        android:id="@+id/menu_feedback"
        android:title="Feedback" />

</menu>
```

res/layout/activity_main.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"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <!-- No UI needed -->  
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.kt

```
package com.example.menuapp  
  
import android.os.Bundle  
import android.view.Menu  
import android.view.MenuItem  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
  
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
  
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {  
        menuInflater.inflate(R.menu.main_menu, menu)  
        return true  
    }  
}
```

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {  
    when (item.itemId) {  
        R.id.menu_settings -> showToast("Settings clicked")  
        R.id.menu_search -> showToast("Search clicked")  
        R.id.menu_feedback -> showToast("Feedback clicked")  
    }  
    return super.onOptionsItemSelected(item)  
}
```

```
private fun showToast(message: String) {  
    Toast.makeText(this, message, Toast.LENGTH_SHORT).show()  
}  
}
```

res/values/themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">  
    <style name="Theme.MenuApp" parent="Theme.AppCompat.Light.DarkActionBar">  
        <item name="colorPrimary">#6200EE</item>  
        <item name="colorPrimaryDark">#3700B3</item>  
        <item name="colorAccent">#03DAC5</item>  
    </style>  
</resources>
```

AndroidManifest.xml

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

```
package="com.example.menuapp">

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="MenuApp"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.MenuApp">

    <activity
        android:name=".MainActivity"
        android:exported="true"><!-- Required for Android 12+ -->
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>

</application>

</manifest>
```

Create an android application to pass the data from current application to another application using intent in android studio

SendDataIntentApp

res/layout/activity_main.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<EditText
    android:id="@+id/inputMessage"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Type your message"
    android:padding="16dp"
    android:layout_marginTop="60dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginHorizontal="16dp"/>
```

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

```
        android:layout_height="wrap_content"
        android:text="Send to Another App"
        app:layout_constraintTop_toBottomOf="@+id/inputMessage"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="24dp"/>

```

</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.kt

```
package com.example.senddataintentapp

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import androidx.appcompat.app.AppCompatActivity

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

        val inputMessage = findViewById<EditText>(R.id.inputMessage)
        val sendButton = findViewById<Button>(R.id.sendButton)

        sendButton.setOnClickListener{

```

```
    val message = inputMessage.text.toString()

    if (message.isNotEmpty()) {
        val intent = Intent(Intent.ACTION_SEND).apply {
            type = "text/plain"
            putExtra(Intent.EXTRA_TEXT, message)
        }

        // Show a chooser so the user can select an app
        val chooser = Intent.createChooser(intent, "Send message with:")
        startActivity(chooser)
    }
}

}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.senddataintentapp">

    <application

        android:allowBackup="true"

        android:label="SendDataIntentApp"

        android:theme="@style/Theme.SendDataIntentApp"

        android:supportsRtl="true"

        android:icon="@mipmap/ic_launcher"
```

```
    android:roundIcon="@mipmap/ic_launcher_round">

<activity
    android:name=".MainActivity"
    android:exported="true"> <!-- Required for Android 12+ -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
</application>

</manifest>
```

Create an android application which automatically notify the user when Aeroplane mode is turned on or off using broadcast receiver on android studio

AirplaneModeNotifier

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.airplanemode notifier">

    <application
        android:allowBackup="true"
        android:label="AirplaneModeNotifier"
        android:supportsRtl="true"
        android:theme="@style/Theme.AirplaneModeNotifier"
        android:icon="@mipmap/ic_launcher"
        android:roundIcon="@mipmap/ic_launcher_round">

        <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>
    </application>
</manifest>
```

```
<!-- Broadcast Receiver Registration -->

<receiver android:name=".AirplaneModeReceiver"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.AIRPLANE_MODE" />
    </intent-filter>
</receiver>
```

```
</application>
```

```
</manifest>
```

MainActivity.kt

```
package com.example.airplanemode notifier

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

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

activity_main.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Toggle Airplane Mode to Test"
    android:textSize="18sp"
    android:layout_marginTop="100dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
app
└── java
    └── com.example.yourapp
        New → Kotlin Class/File
        AirplaneModeReceiver


---


package com.example.airplanemode notifier

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast
```

```
class AirplaneModeReceiver : BroadcastReceiver() {  
    override fun onReceive(context: Context, intent: Intent) {  
        if (intent.action == Intent.ACTION_AIRPLANE_MODE_CHANGED) {  
            val isAirplaneModeOn = intent.getBooleanExtra("state", false)  
            val message = if (isAirplaneModeOn) {  
                " Airplane Mode is ON"  
            } else {  
                " Airplane Mode is OFF"  
            }  
            Toast.makeText(context, message, Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```

