**KOTLIN PROGRAMS:**

1. **.Write a Kotlin program that takes a nullable integer as input and print its Square if it is not null , or “ Input is null” otherwise**

fun main(){

val input:String?= readLine()

val num : Int?=input?.toIntOrNull()

if(num!=null){

println("Square if number : ${num\*num}")

}

else{

println("invalid number ")

}

}

1. **Implement a number guessing game in which the user is prompted to enter a number between 1 and 100 until he or she guesses correctly. After every wrong guess, the user is told whether the guess was too high or too low.**

fun main(){

var r=(1..100).random()

println("Enter the number")

do{

var num:Int=readLine()!!.toInt()

if(num>r){

println("Too high")

}

else if(num<r){

println("Too low")

}

}while(num!=r)

println("Correct number")

}

1. **Create a function in Kotlin that takes a name as input and prints the greeting message. Make the message customizable and provide a default message if no custom message is provided**

fun greet(name:String,msg:String?){

if(msg!=null){

print("Customised message :")

println("hello , $name : $msg")

}

else{

println("$name : $msg")

}

}

fun main(){

print("Enter your name")

var name=readLine()!!.toString()

print("Enter custom message")

var msg = readLine().takeIf{it?.isNotEmpty()==true}

greet(name,msg)

}

1. **Create a program with different types of dwellings (Shelters people live in like roundhut, square cabin, round tower) that are implemented as a class hierarchy**

import kotlin.math.PI

import kotlin.math.sqrt

fun main() {

val squareCabin = SquareCabin(6, 50.0)

val roundHut = RoundHut(3, 10.0)

val roundTower = RoundTower(4, 15.5)

println("\nSquare Cabin\n============")

squareCabin.printDetails()

println("\nRound Hut\n=========")

roundHut.printDetails()

println("Has room? ${roundHut.hasRoom()}")

roundHut.getRoom()

println("Has room? ${roundHut.hasRoom()}")

println("Carpet size: ${roundHut.calculateMaxCarpetLength()}")

println("\nRound Tower\n==========")

roundTower.printDetails()

println("Carpet Length: ${roundTower.calculateMaxCarpetLength()}")

}

// Base class for all dwellings

abstract class Dwelling(private var residents: Int) {

abstract val buildingMaterial: String

abstract val capacity: Int

abstract fun floorArea(): Double

fun hasRoom(): Boolean = residents < capacity

fun getRoom() {

if (hasRoom()) {

residents++

println("You got a room!")

} else {

println("Sorry, no rooms left.")

}

}

fun printDetails() {

println("Material: $buildingMaterial")

println("Capacity: $capacity")

println("Floor area: ${floorArea()}")

}

}

// SquareCabin subclass

class SquareCabin(residents: Int, val length: Double) : Dwelling(residents) {

override val buildingMaterial = "Wood"

override val capacity = 6

override fun floorArea(): Double = length \* length

}

// RoundHut subclass

open class RoundHut(residents: Int, val radius: Double) : Dwelling(residents) {

override val buildingMaterial = "Straw"

override val capacity = 4

override fun floorArea(): Double = PI \* radius \* radius

fun calculateMaxCarpetLength(): Double = sqrt(2.0) \* radius

}

// RoundTower subclass

class RoundTower(residents: Int, radius: Double, val floors: Int = 2) : RoundHut(residents, radius) {

override val buildingMaterial = "Stone"

override val capacity = floors \* 4

override fun floorArea(): Double = super.floorArea() \* floors

}

**Dice Roller app:**

<string name="roll">Roll</string>

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.activity.enableEdgeToEdge

import androidx.compose.foundation.Image

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.Button

import androidx.compose.material3.Text

import androidx.compose.runtime.\*

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.res.stringResource

import androidx.compose.ui.unit.dp

import com.example.dice\_roller.ui.theme.Dice\_RollerTheme

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

enableEdgeToEdge()

setContent {

Dice\_RollerTheme {

DiceRollerApp()

}

}

}

}

@Composable

fun DiceRollerApp() {

DiceWithButtonAndImage(

modifier = Modifier

.fillMaxSize()

.wrapContentSize(Alignment.Center)

)

}

@Composable

fun DiceWithButtonAndImage(modifier: Modifier = Modifier) {

var result by remember { mutableStateOf(1) }

val imageResource = when (result) {

1 -> R.drawable.dice\_1

2 -> R.drawable.dice\_2

3 -> R.drawable.dice\_3

4 -> R.drawable.dice\_4

5 -> R.drawable.dice\_5

else -> R.drawable.dice\_6

}

Column(

modifier = modifier,

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center

) {

Image(

painter = painterResource(imageResource),

contentDescription = "Dice showing $result",

modifier = Modifier.height(180.dp)

)

Spacer(modifier = Modifier.height(16.dp))

Button(onClick = { result = (1..6).random() }) {

Text(stringResource(R.string.roll))

}

}

}

**Create an application to use a single activity and multiple fragments, and navigate between fragments with the Navigation Component.**

package com.example.myapplication7  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.Arrangement  
import androidx.compose.foundation.layout.Column  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.material3.Button  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.navigation.NavController  
import androidx.navigation.compose.NavHost  
import androidx.navigation.compose.composable  
import androidx.navigation.compose.rememberNavController  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** val navController = rememberNavController()  
 NavHost(navController = navController, startDestination = "fragment1") **{** *composable*("fragment1") **{** Fragment1(navController)  
 **}** *composable*("fragment2") **{** Fragment2(navController)  
 **}** *composable*("fragment3") **{** Fragment3(navController)  
 **}  
 }  
 }** }  
}  
  
@Composable  
fun Fragment1(navController: NavController) {  
 Column(  
 modifier = Modifier.*fillMaxSize*(),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text(text = "You are in Fragment 1")  
 Button(onClick = **{** navController.navigate("fragment2")  
 **}**) **{** Text(text = "Go to Fragment 2")  
 **}  
 }**}  
  
@Composable  
fun Fragment2(navController: NavController) {  
 Column(  
 modifier = Modifier.*fillMaxSize*(),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text(text = "You are in Fragment 2")  
 Button(onClick = **{** navController.navigate("fragment3")  
 **}**) **{** Text(text = "Go to Fragment 3")  
 **}  
 }**}  
  
@Composable  
fun Fragment3(navController: NavController) {  
 Column(  
 modifier = Modifier.*fillMaxSize*(),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text(text = "You are in Fragment 3")  
 Button(onClick = **{** navController.navigate("fragment1")  
 **}**) **{** Text(text = "Go to Fragment 1")  
 **}  
 }**}

dependencies add

*implementation*("androidx.navigation:navigation-compose:2.8.3")

**Affirmations**

package com.example.myapplication7  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.Image  
import androidx.compose.foundation.layout.\*  
import androidx.compose.foundation.lazy.LazyColumn  
import androidx.compose.foundation.lazy.items  
import androidx.compose.material3.Card  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.layout.ContentScale  
import androidx.compose.ui.res.painterResource  
import androidx.compose.ui.unit.dp  
  
data class Affirmation(val imageResId: Int, val text: String)  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** AffirmationsApp()  
 **}** }  
}  
  
@Composable  
fun AffirmationsApp() {  
 val affirmations = *listOf*(  
 Affirmation(R.drawable.*news*, "I am a white Cat"),  
 Affirmation(R.drawable.*news*, "I am an Elephant"),  
 Affirmation(R.drawable.*news*, "I am a Tiger"),  
 Affirmation(R.drawable.*news*, "I am a Bird"),  
 Affirmation(R.drawable.*news*, "I am a Fish"),  
 Affirmation(R.drawable.*news*, "I am a Dog")  
 )  
  
 LazyColumn(  
 verticalArrangement = Arrangement.spacedBy(8.*dp*),  
 modifier = Modifier.*padding*(16.*dp*)  
 ) **{** *items*(affirmations) **{** affirmation **->** AffirmationCard(affirmation)  
 **}  
 }**}  
  
@Composable  
fun AffirmationCard(affirmation: Affirmation) {  
 Card(modifier = Modifier.*fillMaxWidth*()) **{** Column **{** Image(  
 painter = painterResource(id = affirmation.imageResId),  
 contentDescription = null,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*height*(200.*dp*),  
 contentScale = ContentScale.Crop  
 )  
 Text(  
 text = affirmation.text,  
 modifier = Modifier.*padding*(16.*dp*)  
 )  
 **}  
 }**}

**Explicit intent**

//MainActivity.kt

package com.example.myapplication  
  
import android.content.Intent  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.activity.enableEdgeToEdge  
import androidx.compose.foundation.layout.Column  
import androidx.compose.foundation.layout.Spacer  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.foundation.layout.fillMaxWidth  
import androidx.compose.foundation.layout.height  
import androidx.compose.foundation.layout.padding  
import androidx.compose.material3.Button  
import androidx.compose.material3.OutlinedTextField  
import androidx.compose.material3.Scaffold  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.getValue  
import androidx.compose.runtime.mutableStateOf  
import androidx.compose.runtime.remember  
import androidx.compose.runtime.setValue  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.platform.*LocalContext*import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import com.example.myapplication.ui.theme.MyApplicationTheme  
  
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        *enableEdgeToEdge*()  
        *setContent* **{**simpleexplicit()  
        **}**}  
}  
  
  
  
@Composable  
fun simpleexplicit(){  
    val context = *LocalContext*.current  
    var uname by remember **{***mutableStateOf*("")  
    **}**Column**{**OutlinedTextField(value = uname, onValueChange = **{**uname=**it}**,  
            label =**{** Text(text = "Enter your name")**}**,  
            modifier =Modifier.*fillMaxWidth*())  
  
        Spacer(modifier =Modifier .*height*(25.*dp*))  
        Button(onClick = **{**val intent = Intent(context,SecondActivity::class.*java*).*apply***{**putExtra("user\_name",uname)  
            **}**context.startActivity(intent)  
        **}**)**{**Text("Send to second Activity")  
        **}  
  
    }**}

//SecondActivity.kt

package com.example.myapplication  
  
import android.annotation.SuppressLint  
import android.os.Bundle  
import android.widget.Toast  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.activity.enableEdgeToEdge  
  
@SuppressLint("RestrictedApi")  
class SecondActivity : ComponentActivity(){  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        val myname = *intent*.getStringExtra("user\_name")  
        Toast.makeText(this,"Welcome $myname",Toast.*LENGTH\_LONG*).show()  
    }  
}

//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">  
  
    <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.MyApplication"  
        tools:targetApi="31">  
        <activity  
            android:name=".MainActivity"  
            android:exported="true"  
            android:label="@string/app\_name"  
            android:theme="@style/Theme.MyApplication">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
        <activity android:name=".SecondActivity"/>  
    </application>  
  
</manifest>

**Room Database Task:**

**User.kt:**

package com.ashwin.insertrecord\_jetpackcompose71024

import androidx.room.Entity

import androidx.room.PrimaryKey

@Entity(tableName = "User")

data class User(

@PrimaryKey(autoGenerate = true)

val uid:Int?=null,

val userName:String,

val userPhone:String

)

**UserDAO.kt:**

import androidx.room.Dao

import androidx.room.Insert

@Dao

interface UserDAO {

@Insert

suspend fun insert(user:User)

}

**UserDatabase.kt:**

import android.content.Context

import androidx.room.Database

import androidx.room.Room

import androidx.room.RoomDatabase

@Database(entities = [User::class], version = 1)

abstract class UserDatabase : RoomDatabase() {

abstract fun userDAO():UserDAO

companion object{

@Volatile

private var INSTANCE:UserDatabase?=null

fun getInstance(context: Context):UserDatabase{

return INSTANCE ?: synchronized(this){

val instance = Room.databaseBuilder(

context.applicationContext,

UserDatabase::class.java,

"SAIPRASAD\_DATABASE"

).build()

INSTANCE=instance

instance

}

}

}

}

**MainActivity.kt:**

import android.os.Bundle

import android.widget.Toast

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.activity.enableEdgeToEdge

import androidx.compose.foundation.layout.Arrangement

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.Spacer

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.height

import androidx.compose.foundation.layout.padding

import androidx.compose.material3.Button

import androidx.compose.material3.OutlinedTextField

import androidx.compose.material3.Scaffold

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.rememberCoroutineScope

import androidx.compose.runtime.setValue

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.platform.LocalContext

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import com.ashwin.insertrecord\_jetpackcompose71024.ui.theme.InsertRecord\_Jetpackcompose71024Theme

import kotlinx.coroutines.CoroutineScope

import kotlinx.coroutines.Dispatchers

//import kotlinx.coroutines.flow.internal.NoOpContinuation.context

import kotlinx.coroutines.launch

//import kotlin.coroutines.jvm.internal.CompletedContinuation.context

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

enableEdgeToEdge()

setContent {

val database = UserDatabase.getInstance(this)

InsertRecord(database)

}

}

}

@Composable

fun InsertRecord(database: UserDatabase){

val context = LocalContext.current

val userdao = database.userDAO()

var name by remember { mutableStateOf("") }

var phone by remember { mutableStateOf("") }

val scope = rememberCoroutineScope()

Column(

modifier = Modifier

.fillMaxSize()

.padding(16.dp),

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Top

) {

Text(text = "Enter Your Details")

Spacer(modifier = Modifier.height(20.dp))

//Name

OutlinedTextField(value = name,

onValueChange = {name = it},

label = {Text("Enter Your Name")},

modifier = Modifier.fillMaxWidth()

)

Spacer(modifier = Modifier.height(8.dp))

//Phone Number

OutlinedTextField(value = phone,

onValueChange = {phone = it},

label = {Text("Enter Your PhoneNumber")},

modifier = Modifier.fillMaxWidth()

)

Spacer(modifier = Modifier.height(16.dp))

Button(onClick = {

//Launch coroutine to insert record

scope.launch(Dispatchers.IO) {

userdao.insert(User(userName=name,

userPhone=phone))

}

Toast.makeText(context, "Record Inserted " +

"Successfully", Toast.LENGTH\_LONG).show()

}) {

Text(text = "Insert Now")

}

}

}

--

plugins **{** alias(*libs*.*plugins*.*android*.*application*)  
 alias(*libs*.*plugins*.*kotlin*.*android*)  
 alias(*libs*.*plugins*.*kotlin*.*compose*)  
 *kotlin*("kapt")  
**}**

**dependencies{**

val room\_version = "2.6.1"  
  
*implementation*("androidx.room:room-runtime:$room\_version")  
  
  
// To use Kotlin annotation processing tool (kapt)  
*kapt*("androidx.room:room-compiler:$room\_version")  
  
  
  
// optional - Kotlin Extensions and Coroutines support for Room  
*implementation*("androidx.room:room-ktx:$room\_version")

}

android developer jetpack compose room database

**HELLO WORLD:**  
  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** Display()  
 **}** }  
}  
  
@Composable  
fun Display() {  
 Column(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*background*(Color.White)  
 .*padding*(16.*dp*), // Optional padding for better spacing  
 horizontalAlignment = Alignment.CenterHorizontally,  
 verticalArrangement = Arrangement.Center  
 ) **{** Text(  
 text = "Hello Sai Prasad",  
 color = Color.Red,  
 fontSize = 35.*sp* )  
 **}**}

**Increment counter:**

package com.example.myapplication7  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.Button  
import androidx.compose.material3.Text  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** Display()  
 **}** }  
}  
  
@Composable  
fun Display() {  
 var counter by remember **{** *mutableStateOf*(0) **}** Column(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*background*(Color.White)  
 .*padding*(16.*dp*),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text(text = "Counter: $counter", color = Color.Red, fontSize = 35.*sp*)  
 Spacer(modifier = Modifier.*height*(16.*dp*))  
 Button(onClick = **{** counter++ **}**) **{** Text(text = "Increment Counter")  
 **}  
 }**}

**both increment and decrement**

import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.Button  
import androidx.compose.material3.Text  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** Display()  
 **}** }  
}  
  
@Composable  
fun Display() {  
 var counter by remember **{** *mutableStateOf*(0) **}** Column(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*background*(Color.White)  
 .*padding*(16.*dp*),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text(text = "Counter: $counter", color = Color.Red, fontSize = 35.*sp*)  
 Spacer(modifier = Modifier.*height*(16.*dp*))  
  
 // Row to place the buttons horizontally  
 Row(  
 horizontalArrangement = Arrangement.spacedBy(16.*dp*)  
 ) **{** Button(onClick = **{** counter++ **}**) **{** Text(text = "Increment")  
 **}** Button(onClick = **{** counter-- **}**) **{** Text(text = "Decrement")  
 **}  
 }  
 }**}

**tip calculator**  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.Button  
import androidx.compose.material3.Text  
import androidx.compose.material3.TextField  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
import com.example.myapplication6.ui.theme.MyApplication6Theme  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** MyApplication6Theme **{** TipCalculator()  
 **}  
 }** }  
}  
  
@Composable  
fun TipCalculator() {  
 var amount by remember **{** *mutableStateOf*("") **}** var tip by remember **{** *mutableStateOf*(0.0) **}** Column(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*background*(Color.White)  
 .*padding*(16.*dp*),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** TextField(  
 value = amount,  
 onValueChange = **{** newValue **->** amount = newValue // Correctly update the amount with new input  
 **}**,  
 label = **{** Text("Enter Amount") **}**,  
 modifier = Modifier.*fillMaxWidth*()  
 )  
  
 Spacer(modifier = Modifier.*height*(16.*dp*))  
  
 Button(onClick = **{** val enteredAmount = amount.*toDoubleOrNull*() ?: 0.0  
 tip = enteredAmount \* 0.15 // Calculate the tip as 15% of the entered amount  
 **}**) **{** Text(text = "Calculate Tip", color = Color.Red, fontSize = 15.*sp*)  
 **}** Spacer(modifier = Modifier.*height*(16.*dp*))  
  
 Text(text = "Tip: $tip", color = Color.Red, fontSize = 35.*sp*)  
 **}**}

**tip user input:**

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.Button

import androidx.compose.material3.Text

import androidx.compose.material3.TextField

import androidx.compose.runtime.\*

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

TipCalculator()

}

}

}

@Composable

fun TipCalculator() {

var amount by remember { mutableStateOf("") }

var tipPercentage by remember { mutableStateOf("") }

var tip by remember { mutableStateOf(0.0) }

Column(

modifier = Modifier

.fillMaxSize()

.background(Color.White)

.padding(16.dp),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

TextField(

value = amount,

onValueChange = { newValue ->

amount = newValue

},

label = { Text("Enter Amount") },

modifier = Modifier.fillMaxWidth()

)

Spacer(modifier = Modifier.height(16.dp))

TextField(

value = tipPercentage,

onValueChange = { newValue ->

tipPercentage = newValue

},

label = { Text("Enter Tip Percentage") },

modifier = Modifier.fillMaxWidth()

)

Spacer(modifier = Modifier.height(16.dp))

Button(onClick = {

val enteredAmount = amount.toDoubleOrNull() ?: 0.0

val enteredTipPercentage = tipPercentage.toDoubleOrNull() ?: 0.0

tip = enteredAmount \* (enteredTipPercentage / 100)

}) {

Text(text = "Calculate Tip", color = Color.Red, fontSize = 15.sp)

}

Spacer(modifier = Modifier.height(16.dp))

Text(text = "Tip: $tip", color = Color.Red, fontSize = 35.sp)

}

}

**ACTIVITY LIFECYCLE:**

// App type: Empty Views Activity

// rename the package name in the below line with your folder name

//MainActivity.kt\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package com.example.lifecycle

import android.os.Bundle

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AppCompatActivity

import android.widget.Toast

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

enableEdgeToEdge()

setContentView(R.layout.activity\_main)

Toast.makeText(applicationContext,"CREATE() CALLED",Toast.LENGTH\_SHORT).show()

}

override fun onStart() {

super.onStart()

Toast.makeText(applicationContext,"START() CALLED",Toast.LENGTH\_SHORT).show()

}

override fun onRestart() {

super.onRestart()

Toast.makeText(applicationContext,"RESTART() CALLED",Toast.LENGTH\_SHORT).show()

}

override fun onPause() {

super.onPause()

Toast.makeText(applicationContext,"PAUSE() CALLED",Toast.LENGTH\_SHORT).show()

}

override fun onResume() {

super.onResume()

Toast.makeText(applicationContext,"RESUME() CALLED",Toast.LENGTH\_SHORT).show()

}

override fun onStop() {

super.onStop()

Toast.makeText(applicationContext,"STOP() CALLED",Toast.LENGTH\_SHORT).show()

}

override fun onDestroy() {

super.onDestroy()

Toast.makeText(applicationContext,"DESTROY() CALLED",Toast.LENGTH\_SHORT).show()

}

}

//\*\*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"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:layout\_width="303dp"

android:layout\_height="66dp"

android:text="ACTIVITY LIFE CYCLE DEMONSTRATION..!"

android:ems="10"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="TextViewEdits"/>

</androidx.constraintlayout.widget.ConstraintLayout>