# Mobile Application Programs

**Program 1 : Creating “Hello world” Application.**

1. activity_main.xml code :

```
<?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">
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click_me"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

2.MainActivity.java Code :

```
package com.example.anushamad03;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button b;
      b=findViewById(R.id.button);
      b.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Toast.makeText(MainActivity.this, "Hey! We are using Android Application",
Toast.LENGTH_SHORT).show();
         }
      });
   }
}
```

**program 2 : Creating an application that displays message based on the screen orientation.**

**1.activity_main.xml code:**

```
<?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">
   <Button
      android:id="@+id/por"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Portrait"
        android:layout_centerInParent="true"/>
    <Button
        android:id="@+id/lan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Landscape"
        android:layout_below="@+id/por"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

**2.MainActivity.java code:**

```
package com.example.anushamad03;
import android.content.pm.ActivityInfo;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button l,p;
        l=findViewById(R.id.lan);
        p=findViewById(R.id.por);
        l.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
                Toast.makeText(MainActivity.this, "Hey! We are in Landscape orientation",
Toast.LENGTH_SHORT).show();
            }
        });
        p.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
                Toast.makeText(MainActivity.this, "Hey! We are in Portrait
orientation",Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

**Program 3 : Create an application that displays custom designed Opening Screen.**

Click Start →Android Studio, a Welcome to Android Studio dialog box will appear.

1. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 (“Jelly Bean”, Android 4.1). Click Finish Button.

**1.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/idTVHeading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_margin="20dp"
        android:gravity="center"
        android:padding="10dp"
        android:text="Background Drawable in Android"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold" />
    </RelativeLayout>
```

**2.Main_activity.java**

```
package com.example.joshimad51;
import android.os.Bundle;
import android.widget.RelativeLayout;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private RelativeLayout containerRL;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        containerRL = findViewById(R.id.main);

        // on below line we are setting background for

        // our relative layout on below line.

        containerRL.setBackground(getResources().getDrawable(R.drawable.back_drawable));

    }

}
```

**Next Step : app --> res--> drawable --> right click on drawable --> Select New --> Click Drawable resource File -->new Resource file window will be openend give filename as "back_drawable.xml" and give root element as shape and set source as main and then click OK.**

back_drawable.xml file is Created

**3.back_drawable.xml Code:**

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

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

    android:shape="rectangle">

        <gradient

            android:angle="270"

            android:endColor="@color/white"

            android:startColor="#2C3A87" />

</shape>
```

## 4. Create a sample application with login module(check user name and password successful login change Textview "Login Successful". On login fail alert using toast "login fail"

**1.Activity_main.xml**

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

   android:layout_centerInParent="true"
android:padding="30dp"

   tools:context=".MainActivity">


   <TextView

       android:layout_width="match_parent"
android:layout_height="48sp"
android:id="@+id/textView3"
android:text="LOGIN PAGE"
android:textAlignment="center"

       android:textSize="30sp" />


   <EditText

       android:layout_width="match_parent"
android:layout_height="48sp"
android:id="@+id/username"

       android:ems="10"
android:inputType="text"

       android:hint="Username" />


   <EditText

       android:id="@+id/password"
android:layout_width="match_pare
nt"
android:layout_height="48sp"
android:ems="10"
android:hint="Password"
android:inputType="text"

       android:password="true" />
   <Button

       android:layout_width="match_parent"
android:layout_height="wrap_content"

       android:id="@+id/login"

       android:text="Login" />

</LinearLayout>
```

**2.MainActivity.java**

```
package com.example.loginmodule;

import android.os.bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import android.widget.EditText;

import android.text.Textutils;

import
androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends
AppCompatActivity {

private EditText inputusername;

private EditText inputPassword;

private button ButtonLogin;

   @Override

   protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);
inputusername= findViewById(R.id.username);
inputPassword =findViewById(R.id.password);
Button Login = findViewById(R.id.button_login);

ButtonLogin.setOnClickListener(new View.OnClickListener() {

 @Override

  public void onClick(View v) {
```

```
String username = inputUsername.getText().toString().trim();
 String password = inputPassword.getText().toString().trim();
if(Textutils.isEmpty(username) || Textutils.isEmpty(Password)){
            Toast.makeText(this, "Username & Password cannot be empty", Toast.LENGTH_SHORT).show();
 } else {
 if (username.equals("admin") && password.equals("1234")) {
                Toast.makeText(MainActivity.this, "Login Succesful!", Toast.LENGTH_SHORT).show();
              }else {
                Toast.makeText(MainActivity.this, "Login fail", Toast.LENGTH_SHORT).show();
              }
           }
       });
 }
```

**Program 5 : Learn to deploy Android applications.**

Steps to Deploy an Android Application

1. Prepare App (use Program 1 Hello world for this program) Optimize performance and test thoroughly.Ensure compatibility with various devices.

**activity_main.xml Code:**

```
<?xml version="1.0" encoding="utf8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/resauto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 tools:context=".MainActivity">

 <TextView
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Hello World!"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 android:textSize="30sp"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

**MainActivity.java**

```
package com.example.helloworld;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity
{
 protected void onCreate(Bundle savedInstanceState)
 {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 }
}
```

2. Generate Signed APK (Android Package Kit):

In Android Studio, navigate to Build > Generate Signed Bundle/APK.

Follow the prompts to create a new keystore or use an existing one. A keystore is a binary file that contains a set of private keys.

Configure the build type (release) and signing configuration.

Generate the signed APK file.

3. Test Your Signed APK:

Before distributing your app, test the signed APK to ensure that the signing process didn't introduce any issues.

Install the APK on various devices and perform thorough testing.

Release on Google Play Console:

Sign in to the Google Play Console (https://play.google.com/apps/publish).

Create a new app entry if this is your first release or select an existing app.

Complete all the required information for the app listing, including the title, description, screenshots, and categorization.

Upload your signed APK file.

Set pricing and distribution options.

Optimize your store listing for search and conversion.

Once everything is set, click the "Publish" button to release your app to the Google Play Store.

5. Other Distribution Channels (Optional):

Besides Google Play, you can distribute your app through other channels such as Amazon Appstore, Samsung Galaxy Store, or third party app marketplaces.

Each distribution channel may have its own requirements and submission process, so be sure to follow their guidelines.

6. Monitor and Update:

Keep an eye on user feedback and app performance metrics through the Google Play Console.

Regularly update your app to fix bugs, add new features, and improve user experience based on feedback