Program 1:

1. Creating “Hello world” Application.

1. Click Start →Android Studio, a Welcome to Android Studio dialog box will appear.
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.
4. Create a Button resource in activity_main.xml and update the following code.

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/hello"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:background="#535538"
 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>
```

MainActivity.java Code :

Create a Button object, create clickListener, onClick event and update the following code in MainActivity.java

```
package com.example.hello_world;
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.hello);
            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:

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

1. Click Start →Android Studio, a Welcome to Android Studio dialog box will appear.
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.
4. Create two Button resources in activity_main.xml and update the following code.

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">

    <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>
```

5. Create two Button object, create clickListener, onClick event and update the following code in MainActivity.java

MainActivity.Java code :

```
package com.example.joshimad1;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        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:

3. Create an application to develop Login window using UI controls.

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).
4. Click Finish Button.
5. Create two EditText box and a Button resource in activity_main.xml and update the following code.

activity_main.xml code:

```
<?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">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_centerInParent="true"
        android:orientation="vertical"
        android:padding="30dp"
        >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="LOGIN PAGE"
            android:textSize="32sp"
            android:textStyle="bold"
            android:fontFamily="sans-serif-condensed-medium"
            android:textColor="@color/black"
            android:paddingBottom="20dp"
            />
        <EditText
```

```
            android:id="@+id/editTextUsername"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Username"
            android:layout_marginBottom="16dp"/>
        <EditText
            android:id="@+id/editTextPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:layout_below="@id/editTextUsername"
            android:layout_marginBottom="16dp"
            android:inputType="textPassword"/>
        <Button
            android:id="@+id/buttonLogin"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Login"
            android:layout_below="@id/editTextPassword"/>
    </LinearLayout>
</RelativeLayout>
```

6. Create two EditText and a Button object, create clickListener, onClick event for
button object and update the following code in MainActivity.java

MainActivity.Java code:

```
package com.example.joshimad21;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private EditText editTextUsername,editTextPassword;
    private Button buttonLogin;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        editTextUsername = findViewById(R.id.editTextUsername);
        editTextPassword = findViewById(R.id.editTextPassword);
        buttonLogin = findViewById(R.id.buttonLogin);
        buttonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = editTextUsername.getText().toString().trim();
                String password = editTextPassword.getText().toString().trim();
                if(username.equals("admin") && password.equals("pass")){
                    Toast.makeText(MainActivity.this, "Login successful",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Invalid username or
password", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

Program 4 :

4. Create an application to implement new activity using explicit intent, implicit intent and content provider. 5

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).
4. Click Finish Button.

activity_main.xml code:

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:padding="30dp">
    <Button
        android:id="@+id/btnExplicitContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Explicit Content"
        android:textSize="30sp"
        android:layout_marginTop="30dp"></Button>

</LinearLayout>
```

Mainactivity.Java code:

```
package com.example.joshimad31;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    Button btnExplicitContent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnExplicitContent=findViewById(R.id.btnExplicitContent);
        btnExplicitContent.setOnClickListener(new View.OnClickListener()
{ @Override
        public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, NewActivity.class);
            startActivity(intent);
        }
        });
    }
}
```

Next step : To create another activity for Explicit Intent, Go to File --> Click On New --> window opens select Activity ---> Empty Views Activity

opens a dialog window Empty Views activity there give file name as NextActivity and Layout name is activity_new click ok

activity_new.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"
    tools:context=".NewActivity"
    android:orientation="vertical"
    android:padding="30dp">
    <Button
        android:id="@+id/btnImplicitContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Implicit Content"
        android:textSize="30sp"
        android:layout_marginTop="30dp">

    </Button>
</LinearLayout>
```

Newactivity.java code :

```
package com.example.joshimad31;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class NewActivity extends AppCompatActivity {
    Button btnImplicitContent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_new);
        btnImplicitContent=findViewById(R.id.btnImplicitContent);
        btnImplicitContent.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Uri webpage = Uri.parse("http://www.openglprojects.in");
                Intent intent = new Intent(Intent.ACTION_VIEW, webpage);
                startActivity(intent);
            }
        });
    }
}
```

Program 5:

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.

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>
```

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

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>
```

Program Mad 6:

Create an application to perform button Click Event

activity_main.xml code:

```
<?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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, KLESNC BCA STUDENTS"
        android:textSize="24sp"
        android:layout_centerInParent="true" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me"
        android:layout_below="@id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp" />

</RelativeLayout>
```

MainActivity.Java code:

```
package com.example.joshi_mad6;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private TextView textView;
    private Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize views
        textView = findViewById(R.id.textView);
        button = findViewById(R.id.button);

        // Set click listener for the button
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                textView.setText("All the best");
            }
        });
    }
}
```

Program Mad 7:

Create an UI with all views

activity_main code:

```
<?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">

    <!-- Name EditText -->
    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Name"
        android:inputType="text"/>

    <!-- Date of Birth EditText -->
    <EditText
        android:id="@+id/editTextDOB"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Date of Birth"
        android:inputType="date"/>

    <!-- Gender RadioGroup -->
    <RadioGroup
        android:id="@+id/radioGroupGender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <RadioButton
            android:id="@+id/radioButtonMale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Male"/>

        <RadioButton
            android:id="@+id/radioButtonFemale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Female"/>

    </RadioGroup>

    <!-- Checkbox for terms and conditions -->
    <CheckBox
        android:id="@+id/checkBoxTerms"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I agree to the Terms and Conditions"/>

    <!-- Submit Button -->
    <Button
        android:id="@+id/buttonSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Register"/>

</LinearLayout>
```

MainActivity.java code :

```
package com.example.joshi_mad_7;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText editTextName, editTextDOB;
    private RadioGroup radioGroupGender;
    private CheckBox checkBoxTerms;
    private Button buttonSubmit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize views
        editTextName = findViewById(R.id.editTextName);
        editTextDOB = findViewById(R.id.editTextDOB);
        radioGroupGender = findViewById(R.id.radioGroupGender);
        checkBoxTerms = findViewById(R.id.checkBoxTerms);
        buttonSubmit = findViewById(R.id.buttonSubmit);

        // Set click listener for the submit button
        buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Retrieve user input
                String name = editTextName.getText().toString().trim();
                String dob = editTextDOB.getText().toString().trim();
                int selectedGenderId =
radioGroupGender.getCheckedRadioButtonId();
                String gender = getGenderFromId(selectedGenderId);
                boolean termsChecked = checkBoxTerms.isChecked();

                // Validate user input
                if (name.isEmpty() || dob.isEmpty() || gender.isEmpty() || !
termsChecked) {
                    Toast.makeText(MainActivity.this, "Please fill in all fields
and agree to terms.", Toast.LENGTH_SHORT).show();
                } else {
                    // Process registration (e.g., save to database, etc.)
                    Toast.makeText(MainActivity.this, "Registration
successful!", Toast.LENGTH_SHORT).show();
                    // Optionally, you can navigate to another activity or
perform other actions here
                }
            }
        });
    }
```

```
    // Helper method to get gender from radio button ID
    private String getGenderFromId(int selectedId) {
        RadioButton radioButtonMale = findViewById(R.id.radioButtonMale);
        RadioButton radioButtonFemale = findViewById(R.id.radioButtonFemale);

        if (selectedId == radioButtonMale.getId()) {
            return "Male";
        } else if (selectedId == radioButtonFemale.getId()) {
            return "Female";
        } else {
            return "";
        }
    }
}
```

Program 8 : Learn to deploy Android applications.

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