MOBILE APPLICATION DEVELOPMENT LAB

CYCLE 3

1. **Simple login page using Relative Layout**

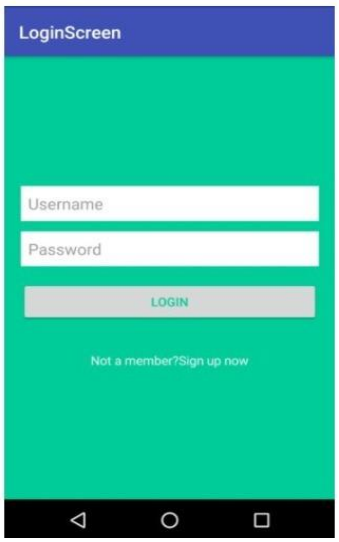
Activity_main.xml

```
<?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:id="@+id/activity_main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="#00CC99">


   <EditText
      android:id="@+id/text1"
      android:hint="Username"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="150dp"
      android:layout_marginLeft="18dp"
      android:layout_marginRight="18dp"
      android:padding="8dp"
      android:background="#fff" />

   <EditText
      android:id="@+id/text2"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginLeft="18dp"
      android:layout_marginRight="18dp"
      android:padding="8dp"
      android:background="#fff"
      android:hint="Password"
      android:layout_marginTop="12dp"
      android:layout_below="@+id/text1" />

   <Button
      android:id="@+id/b1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Login"
      android:textColor="#00CC99"
      android:layout_below="@+id/text2"
      android:layout_marginTop="17dp"
      android:layout_alignStart="@+id/text2"
```

```
        android:layout_alignEnd="@+id/text2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/text3"
        android:textColor="#fff"
        android:text="Not a member?Sign up now"
        android:layout_below="@+id/b1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="34dp" />
</RelativeLayout>
```

MainActivity.java

```
package com.codedost.loginscreen;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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



2. **Array Adapter with list view**

activity_main.xml

```
<?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">

    <ListView
        android:id="@+id/simpleListView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

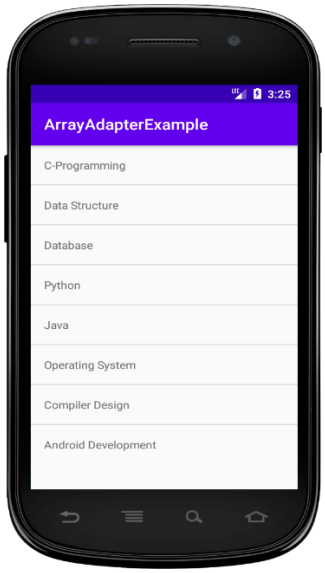
item_view.xml

```
<?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">

    <TextView
        android:id="@+id/itemTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

</LinearLayout>
```

MainActivity.java

```
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
```

```
    ListView simpleListView;

    // array objects
    String courseList[] = {"C-Programming", "Data Structure", "Database", "Python",
                                                "Java", "Operating System", "Compiler
Design", "Android Development"};

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

        simpleListView = (ListView) findViewById(R.id.simpleListView);

        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
                    R.layout.item_view, R.id.itemTextView, courseList);
        simpleListView.setAdapter(arrayAdapter);
    }
}
```



3. Develop an application that toggle image using frame format

Activity_main.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">

 <ImageView
   android:id="@+id/imageview"
   android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent"
    android:scaleType="fitCenter"
    android:src="@drawable/piq1" />

<Button
    android:id="@+id/next"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_marginBottom="15dp"
    android:layout_marginRight="10dp"
    android:layout_gravity="bottom|right"
    android:paddingTop="2dp"
    android:paddingBottom="2dp"
    android:background="@drawable/buttonback"
    android:textColor="#000000"
    android:text="Next" />

  </FrameLayout>
```

MainActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class Piqlout extends Activity {

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.piq);

    Button next= (Button) findViewById(R.id.next);

    if (next.getText().equals("Next")) {
    next.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            ImageView img = (ImageView) findViewById(R.id.imageview);
```

```
            img.setImageResource(R.drawable.piq2);
            Button next= (Button) findViewById(R.id.next);
            next.setText("Prev");
        }
    });
    }
    if (next.getText().equals("Prev")){
    next.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub


                ImageView img = (ImageView)     findViewById(R.id.imageview);
                img.setImageResource(R.drawable.piq1);
                Button next= (Button) findViewById(R.id.next);
                next.setText("Next");
        }
    });
    }
}
```

4. Demonstrate Activity Life Cycle

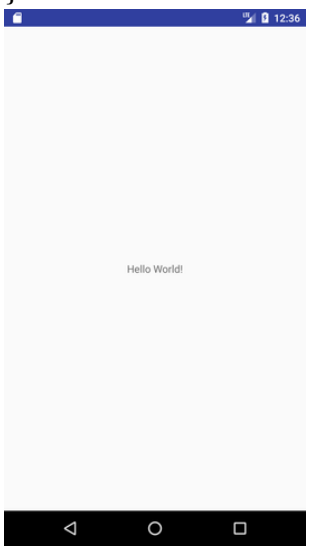
Activity_main.xml

```
<?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"
    tools:context="example.javatpoint.com.activitylifecycle.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

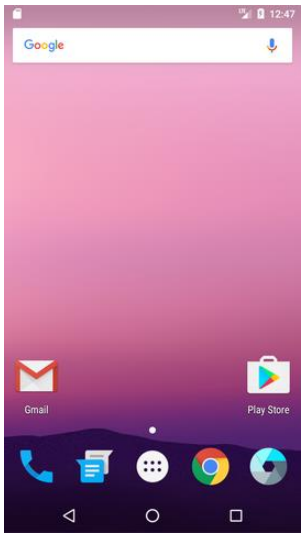
MainActivity.java

```
package example.javatpoint.com.activitylifecycle;
import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle","onCreate invoked");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle","onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle","onResume invoked");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle","onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle","onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("lifecycle","onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle","onDestroy invoked");
    }
```

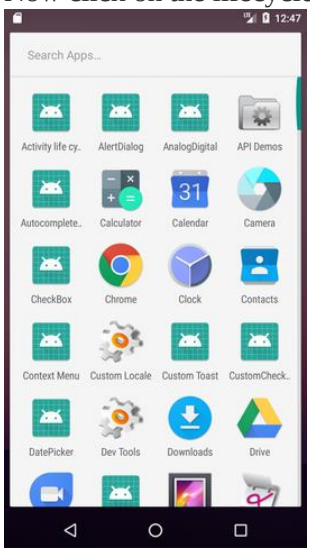
}



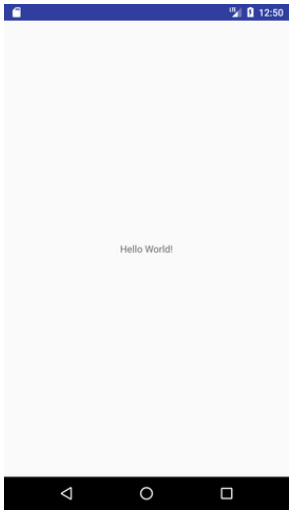
onCreate, onStart and onResume methods are invoked.
Now click on the HOME Button. You will see onPause method is invoked.
Now see on the emulator. It is on the home. Now click on the center button to launch the app again.



Now click on the lifecycleactivity icon.



onRestart, onStart and onResume methods are invoked.



Now click on the back button. Now you will see onPause methods is invoked.
After a while, you will see onStop and onDestroy methods are invoked.

5. Taking camera and saving the picture

Mainactivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="https://schemas.android.com/apk/res/android"
    xmlns:app="https://schemas.android.com/apk/res-auto"
    xmlns:tools="https://schemas.android.com/tools"
    android:id="@+id/content_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="#000000"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.journaldev.imagepicker.MainActivity"
    tools:showIn="@layout/activity_main">
```

```
<RelativeLayout
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:background="@drawable/image_border"
    android:clickable="true"
    android:orientation="vertical">


    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:adjustViewBounds="true"
        android:scaleType="centerCrop" />

</RelativeLayout>

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/img_profile"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="center_horizontal"
    android:src="@drawable/profile"
    app:civ_border_width="5dp"
    app:civ_border_color="#FFFFFF"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true" />
```

```
</RelativeLayout>
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    Bitmap myBitmap;
    Uri picUri;

    private ArrayList permissionsToRequest;
    private ArrayList permissionsRejected = new ArrayList();
    private ArrayList permissions = new ArrayList();

    private final static int ALL_PERMISSIONS_RESULT = 107;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivityForResult(getPickImageChooserIntent(), 200);
            }
        });
```

```
        permissions.add(CAMERA);
        permissionsToRequest = findUnAskedPermissions(permissions);
        //get the permissions we have asked for before but are not granted..
        //we will store this in a global list to access later.

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

            if (permissionsToRequest.size() > 0)
                requestPermissions(permissionsToRequest.toArray(new
String[permissionsToRequest.size()]), ALL_PERMISSIONS_RESULT);
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }
```

```
        return super.onOptionsItemSelected(item);
    }
    /**
     * Create a chooser intent to select the source to get image from.<br />
     * The source can be camera's (ACTION_IMAGE_CAPTURE) or gallery's (ACTION_GET_CONTENT).<br />
     * All possible sources are added to the intent chooser.
     */
    public Intent getPickImageChooserIntent() {

        // Determine Uri of camera image to save.
        Uri outputFileUri = getCaptureImageOutputUri();

        List allIntents = new ArrayList();
        PackageManager packageManager = getPackageManager();

        // collect all camera intents
        Intent captureIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        List listCam = packageManager.queryIntentActivities(captureIntent, 0);
        for (ResolveInfo res : listCam) {
            Intent intent = new Intent(captureIntent);
            intent.setComponent(new ComponentName(res.activityInfo.packageName, res.activityInfo.name));
            intent.setPackage(res.activityInfo.packageName);
            if (outputFileUri != null) {
                intent.putExtra(MediaStore.EXTRA_OUTPUT, outputFileUri);
            }
            allIntents.add(intent);
        }
```

```
        // collect all gallery intents
        Intent galleryIntent = new Intent(Intent.ACTION_GET_CONTENT);
        galleryIntent.setType("image/*");
        List listGallery = packageManager.queryIntentActivities(galleryIntent, 0);
        for (ResolveInfo res : listGallery) {
            Intent intent = new Intent(galleryIntent);
            intent.setComponent(new ComponentName(res.activityInfo.packageName,
res.activityInfo.name));
            intent.setPackage(res.activityInfo.packageName);
            allIntents.add(intent);
        }

        // the main intent is the last in the list (fucking android) so pickup the useless one
        Intent mainIntent = allIntents.get(allIntents.size() - 1);
        for (Intent intent : allIntents) {
            if
(intent.getComponent().getClassName().equals("com.android.documentsui.DocumentsActivi
ty")) {
                mainIntent = intent;
                break;
            }
        }
        allIntents.remove(mainIntent);

        // Create a chooser from the main intent
        Intent chooserIntent = Intent.createChooser(mainIntent, "Select source");

        // Add all other intents
        chooserIntent.putExtra(Intent.EXTRA_INITIAL_INTENTS, allIntents.toArray(new
Parcelable[allIntents.size()]));

        return chooserIntent;
```

```
    }
    /**
     * Get URI to image received from capture by camera.
     */
    private Uri getCaptureImageOutputUri() {
        Uri outputFileUri = null;
        File getImage = getExternalCacheDir();
        if (getImage != null) {
            outputFileUri = Uri.fromFile(new File(getImage.getPath(), "profile.png"));
        }
        return outputFileUri;
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        Bitmap bitmap;
        if (resultCode == Activity.RESULT_OK) {

            ImageView imageView = (ImageView) findViewById(R.id.imageView);

            if (getPickImageResultUri(data) != null) {
                picUri = getPickImageResultUri(data);

                try {
                    myBitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(),
picUri);
                    myBitmap = rotateImageIfRequired(myBitmap, picUri);
                    myBitmap = getResizedBitmap(myBitmap, 500);

                    CircleImageView croppedImageView = (CircleImageView)
findViewById(R.id.img_profile);
```

```
                croppedImageView.setImageBitmap(myBitmap);
                imageView.setImageBitmap(myBitmap);

            } catch (IOException e) {
                e.printStackTrace();
            }

        } else {

            bitmap = (Bitmap) data.getExtras().get("data");

            myBitmap = bitmap;
            CircleImageView croppedImageView = (CircleImageView)
findViewById(R.id.img_profile);
            if (croppedImageView != null) {
                croppedImageView.setImageBitmap(myBitmap);
            }
            imageView.setImageBitmap(myBitmap);

        }
    }
}

  private static Bitmap rotateImageIfRequired(Bitmap img, Uri selectedImage) throws
IOException {

    ExifInterface ei = new ExifInterface(selectedImage.getPath());
    int orientation = ei.getAttributeInt(ExifInterface.TAG_ORIENTATION,
ExifInterface.ORIENTATION_NORMAL);

    switch (orientation) {
```

```
            case ExifInterface.ORIENTATION_ROTATE_90:
                return rotateImage(img, 90);
            case ExifInterface.ORIENTATION_ROTATE_180:
                return rotateImage(img, 180);
            case ExifInterface.ORIENTATION_ROTATE_270:
                return rotateImage(img, 270);
            default:
                return img;
        }
    }

    private static Bitmap rotateImage(Bitmap img, int degree) {
        Matrix matrix = new Matrix();
        matrix.postRotate(degree);
        Bitmap rotatedImg = Bitmap.createBitmap(img, 0, 0, img.getWidth(), img.getHeight(),
matrix, true);
        img.recycle();
        return rotatedImg;
    }

    public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
        int width = image.getWidth();
        int height = image.getHeight();

        float bitmapRatio = (float) width / (float) height;
        if (bitmapRatio > 0) {
            width = maxSize;
            height = (int) (width / bitmapRatio);
        } else {
            height = maxSize;
            width = (int) (height * bitmapRatio);
```

```
        }
        return Bitmap.createScaledBitmap(image, width, height, true);
    }
    /**
     * Get the URI of the selected image from {@link #getPickImageChooserIntent()}.<br />
     * Will return the correct URI for camera and gallery image.
     *
     * @param data the returned data of the activity result
     */
    public Uri getPickImageResultUri(Intent data) {
        boolean isCamera = true;
        if (data != null) {
            String action = data.getAction();
            isCamera = action != null &&
action.equals(MediaStore.ACTION_IMAGE_CAPTURE);
        }
        return isCamera ? getCaptureImageOutputUri() : data.getData();
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);

        // save file url in bundle as it will be null on scren orientation
        // changes
        outState.putParcelable("pic_uri", picUri);
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
```

```
        // get the file url
        picUri = savedInstanceState.getParcelable("pic_uri");
    }

    private ArrayList findUnAskedPermissions(ArrayList wanted) {
        ArrayList result = new ArrayList();

        for (String perm : wanted) {
            if (!hasPermission(perm)) {
                result.add(perm);
            }
        }

        return result;
    }

    private boolean hasPermission(String permission) {
        if (canMakeSmores()) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                return (checkSelfPermission(permission) ==
PackageManager.PERMISSION_GRANTED);
            }
        }
        return true;
    }

    private void showMessageOKCancel(String message, DialogInterface.OnClickListener
okListener) {
        new AlertDialog.Builder(this)
                .setMessage(message)
```

```
            .setPositiveButton("OK", okListener)
            .setNegativeButton("Cancel", null)
            .create()
            .show();
    }
    private boolean canMakeSmores() {
        return (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP_MR1);
    }

    @TargetApi(Build.VERSION_CODES.M)
    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {

        switch (requestCode) {

            case ALL_PERMISSIONS_RESULT:
                for (String perms : permissionsToRequest) {
                    if (hasPermission(perms)) {

                    } else {

                        permissionsRejected.add(perms);
                    }
                }

                if (permissionsRejected.size() > 0) {


                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                        if (shouldShowRequestPermissionRationale(permissionsRejected.get(0))) {
```

```
                    showMessageOKCancel("These permissions are mandatory for the
application. Please allow access.",
                            new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
{
                                        //Log.d("API123", "permisionrejected " +
permissionsRejected.size());
                                        requestPermissions(permissionsRejected.toArray(new
String[permissionsRejected.size()]), ALL_PERMISSIONS_RESULT);
                                    }
                                }
                            });
                    return;
                }
            }
        }
        break;
    }
  }
}
```