Assignment - 1

18CP047 (Pavan Gabani)

1. **Write an android program to send data like username, password, email and contact no. from one activity to another activity using explicit intent.**

AndroidMenifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.paratparinfotech. datapassingtwoactivity">
<uses-permission android:name="android.permission.INTERNET" />
   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme.DatePicker">
      <activity android:name=".MainActivity">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>
</manifest>
```

Activities

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:layout_width="match_parent"
android:orientation="vertical"
android:padding="8dp"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Registration Details"
   android:layout_gravity="center_horizontal"
   android:textSize="30sp"
   android:textColor="@color/black"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintLeft_toLeftOf="parent"
   app:layout_constraintRight_toRightOf="parent"
   app:layout_constraintTop_toTopOf="parent" />
<EditText
   android:id="@+id/username"
   android:layout_width="match_parent"
   android:hint="Username"
   android:layout_height="wrap_content"
   />
<EditText
   android:id="@+id/email"
   android:layout_width="match_parent"
   android:hint="Email"
   android:layout_height="wrap_content"
   />
<EditText
   android:id="@+id/password"
   android:layout_width="match_parent"
```

```
    android:hint="Password"
    android:layout_height="wrap_content"
    />
  <EditText
    android:id="@+id/contact"
    android:layout_width="match_parent"
    android:hint="Contact Number"
    android:layout_height="wrap_content"
    />
  <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/submit"
    android:text="Submit"/>
</LinearLayout>
```

second_activity.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:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".SecondActivity">
  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textSize="20sp"
```

```
        android:textColor="@color/black"
        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

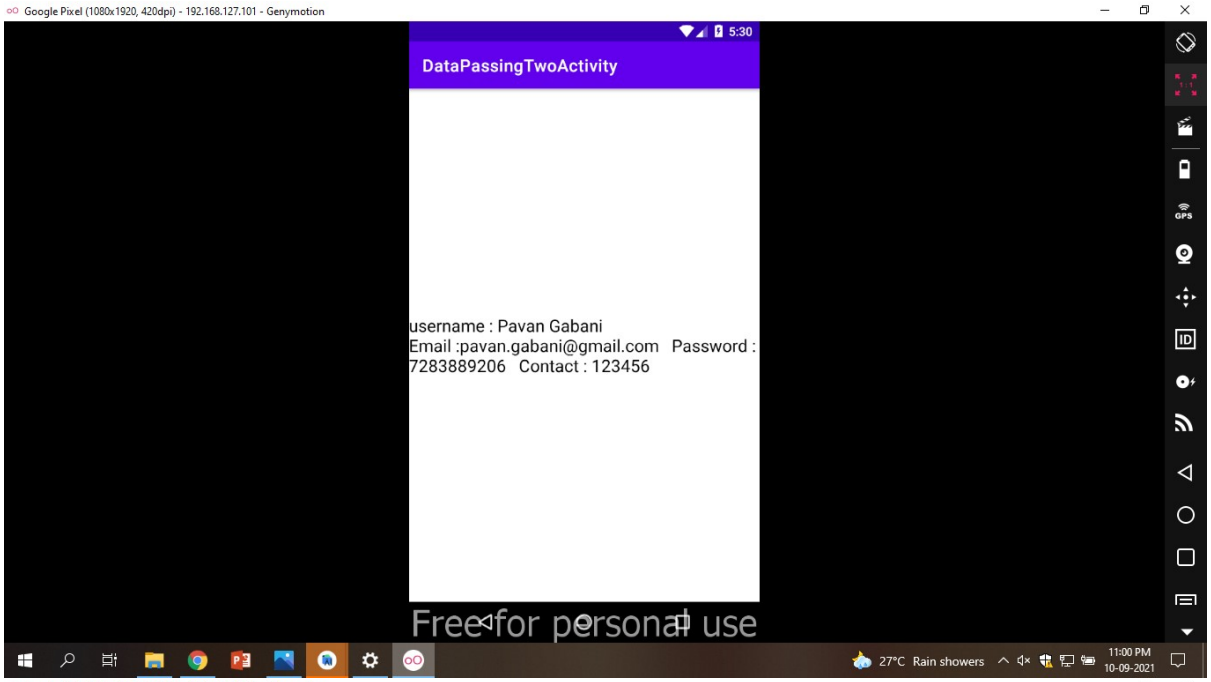
```
package com.paratparinfotech.datapassingtwoactivity;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.paratparinfotech.datapassingtwoactivity.databinding.ActivityMainBinding;
public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;
    EditText username,password,email,contact;
    Button submit;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        View view = binding.getRoot();
        setContentView(view);
        binding.submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this,SecondActivity.class);
                intent.putExtra("username",binding.username.getText().toString());
```

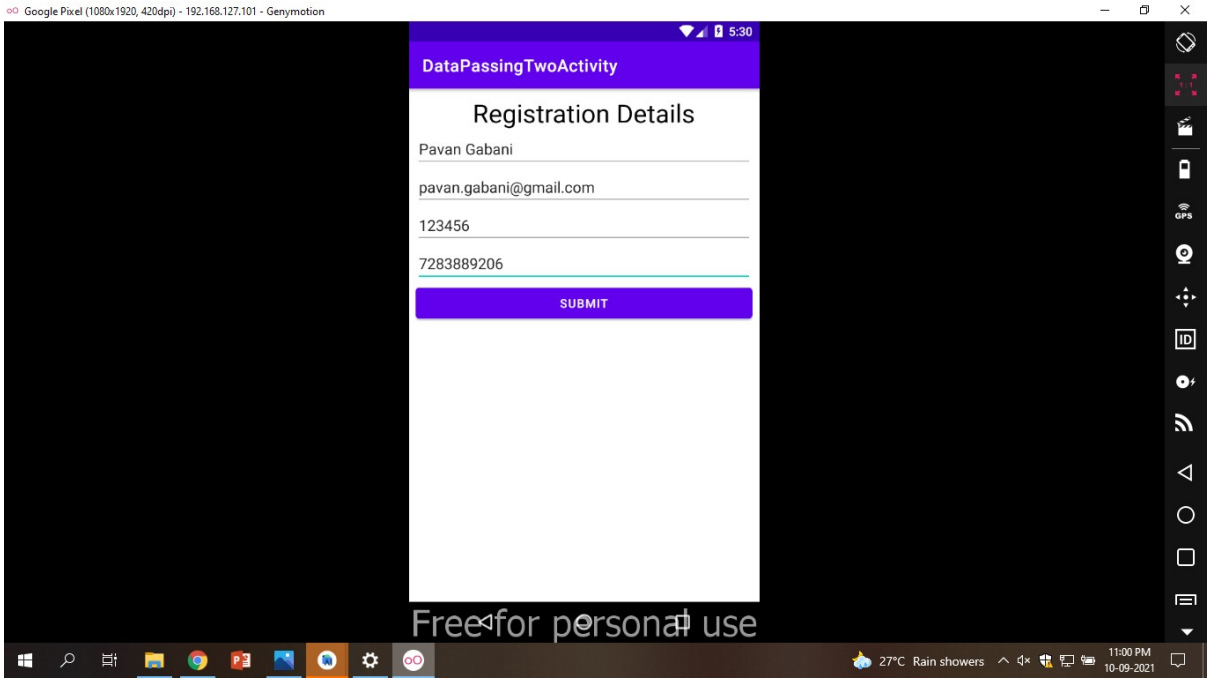
```
            intent.putExtra("email",binding.email.getText().toString());
            intent.putExtra("contact",binding.contact.getText().toString());
            intent.putExtra("password",binding.password.getText().toString());
            startActivity(intent);
        }
      });
   }
}
```

SecondActivity.java

```
package com.paratparinfotech.datapassingtwoactivity;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
public class SecondActivity extends AppCompatActivity {
   TextView text;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_second);
      text = findViewById(R.id.textView);

      Intent intent = getIntent();
      String s = intent.getStringExtra("username") + intent.getStringExtra("email") +
            intent.getStringExtra("contact") + intent.getStringExtra("password");
      text.setText(s);
   }
}
```

Output :


Google Pixel (1080x1920, 420dpi) - 192.168.127.101 - Genymotion
5:30
DataPassingTwoActivity
username : Pavan Gabani
Email :pavan.gabani@gmail.com Password :
7283889206 Contact : 123456
Free for personal use
27°C Rain showers
11:00 PM
10-09-2021


Google Pixel (1080x1920, 420dpi) - 192.168.127.101 - Genymotion
5:30
DataPassingTwoActivity
Registration Details
Pavan Gabani
pavan.gabani@gmail.com
123456
7283889206
SUBMIT
Free for personal use
27°C Rain showers
11:00 PM
10-09-2021

2. **Write a program to add ProgressDialog in android application and sent the result to next activity using implicit intent. (Progress in range of 1 to 200)**

MainActivity.java

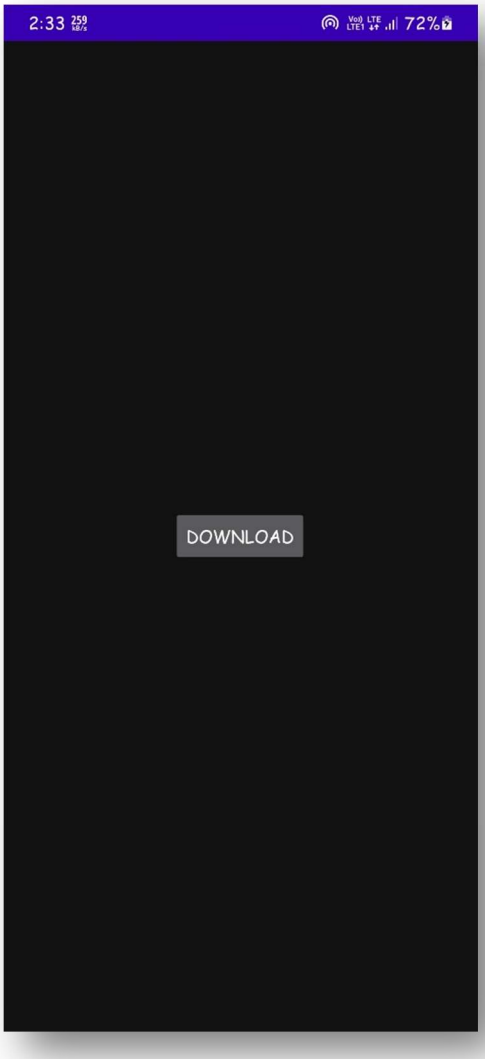
```
package com.paratparinfotech.progressdialog;
import android.app.Activity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
public class MainActivity extends Activity {
   Button btnStartProgress;
   ProgressDialog progressBar;
   private int progressBarStatus = 0;
   private Handler progressBarHandler = new Handler();
   private long fileSize = 0;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      addListenerOnButtonClick();
   }
   public void addListenerOnButtonClick() {
      btnStartProgress = findViewById(R.id.button);
      btnStartProgress.setOnClickListener(new View.OnClickListener(){

         @Override
         public void onClick(View v) {
            // creating progress bar dialog
            progressBar = new ProgressDialog(v.getContext());
            progressBar.setCancelable(true);
            progressBar.setMessage("File downloading ...");
            progressBar.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            progressBar.setProgress(0);
            progressBar.setMax(200);
            progressBar.show();
            //reset progress bar and filesize status
            progressBarStatus = 0;
            fileSize = 0;

            new Thread(new Runnable() {
               public void run() {
                  while (progressBarStatus < 200) {
                     // performing operation
                     progressBarStatus = doOperation();
                     try {
```

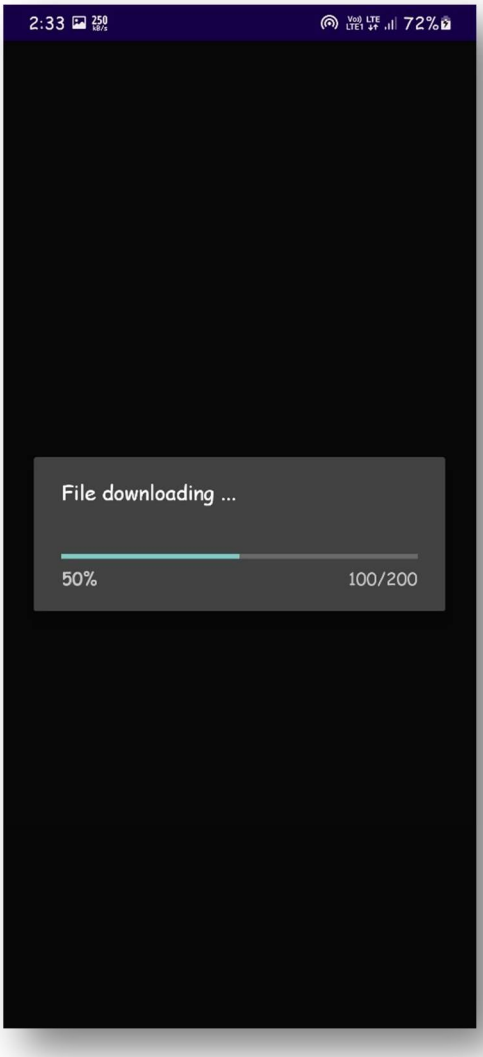
```
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                // Updating the progress bar
                progressBarHandler.post(new Runnable() {
                    public void run() {
                        progressBar.setProgress(progressBarStatus);
                    }
                });
            }
            // performing operation if file is downloaded,
            if (progressBarStatus >= 200) {
                // sleeping for 1 second after operation completed
                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                // close the progress bar dialog
                progressBar.dismiss();
            }
        }
      }).start();
    }//end of onClick method
  });
 }
 // checking how much file is downloaded and updating the filesize
 public int doOperation() {
    //The range of ProgressDialog starts from 0 to 10000
    while (fileSize <= 10000) {
      fileSize++;
      if (fileSize == 1000) {
        return 10;
      } else if (fileSize == 2000) {
        return 20;
      } else if (fileSize == 3000) {
        return 30;
      } else if (fileSize == 4000) {
        return 40; // you can add more else if
      }
     /* else {
        return 100;
      }*/
    }//end of while
    return 100;
 }//end of doOperation
}
```

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: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="Download"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

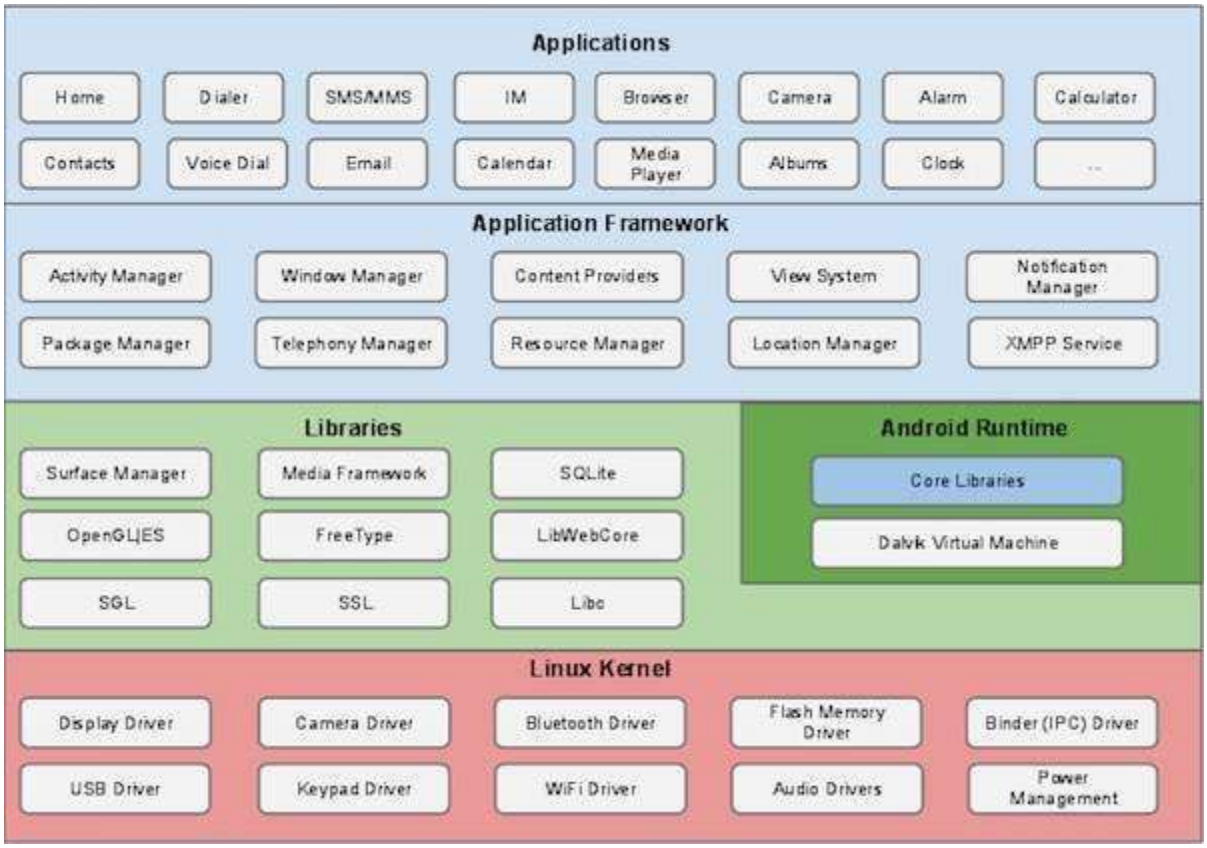
Output :


2:33
72%
DOWNLOAD


2:33
72%
File downloading ...
50%
100/200

3. Explain Android Architecture in details with suitable example.

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



Linux kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows −

android.app − Provides access to the application model and is the cornerstone of all Android applications.

android.content − Facilitates content access, publishing and messaging between applications and application components.

android.database − Used to access data published by content providers and includes SQLite database management classes.

android.opengl − A Java interface to the OpenGL ES 3D graphics rendering API.

android.os − Provides applications with access to standard operating system services including messages, system services and inter-process communication.

android.text − Used to render and manipulate text on a device display.

android.view − The fundamental building blocks of application user interfaces.

android.widget − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.

android.webkit − A set of classes intended to allow web-browsing capabilities to be built into applications.

Having covered the Java-based core libraries in the Android runtime, it is now time to turn our attention to the C/C++ based libraries contained in this layer of the Android software stack.

Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services −

Activity Manager − Controls all aspects of the application lifecycle and activity stack.

Content Providers − Allows applications to publish and share data with other applications.

Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.

Notifications Manager − Allows applications to display alerts and notifications to the user.

View System − An extensible set of views used to create application user interfaces.

Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

4. **Create an android application that demonstrate simple calculator application.**

AndroidMenifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.paratparinfotech.calculator">
   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme.Calculator">
      <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>
```

MainActivity.java

```
package com.paratparinfotech.calculator;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
import com.paratparinfotech.calculator.databinding.ActivityMainBinding;
import java.util.concurrent.ExecutionException;
import javax.xml.xpath.XPathExpression;
public class MainActivity extends AppCompatActivity {
   private ActivityMainBinding binding;
   private StringBuffer equation;
   private StringBuffer ans;
   @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityMainBinding.inflate(getLayoutInflater());
    View view = binding.getRoot();
    setContentView(view);
    equation = new StringBuffer();
    ans = new StringBuffer();
    binding.clear.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            doClear();
        }
    });
    binding.delete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(equation.length()==0) return ;
            equation.delete(equation.length()-1,equation.length());
            setValues();
        }
    });
    binding.zero.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append("0");
            setValues();
        }
    });
    binding.one.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append("1");
            setValues();
        }
    });
    binding.two.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append("2");
            setValues();
        }
    });
    binding.three.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append("3");
            setValues();
        }
```

```
});
binding.four.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append("4");
        setValues();
    }
});
binding.five.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append("5");
        setValues();
    }
});
binding.six.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append("6");
        setValues();
    }
});
binding.seven.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append("7");
        setValues();
    }
});
binding.eight.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append("8");
        setValues();
    }
});
binding.nine.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append("9");
        setValues();
    }
});
binding.dot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        equation.append(".");
        setValues();
```

```
        }
    });
    binding.equal.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                ans = new StringBuffer(String.valueOf(EvaluateString.evaluate(equation.toString())));
//                Toast.makeText(getApplicationContext(),
""+EvaluateString.evaluate(equation.toString()), Toast.LENGTH_SHORT).show();
                equation = new StringBuffer("");
            }
            catch (Exception e){
                Toast.makeText(getApplicationContext(), ""+e.getMessage(),
Toast.LENGTH_SHORT).show();
            }


            setValues();
        }
    });
    binding.add.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append(" + ");
            setValues();
        }
    });
    binding.sub.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append(" - ");
            setValues();
        }
    });
    binding.multiply.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append(" * ");
            setValues();
        }
    });
    binding.divide.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            equation.append(" / ");
            setValues();
        }
    });
```

```
    }

    void doClear(){
        equation.delete(0, equation.length());
        ans.delete(0, ans.length());
        setValues();
    }
    void setValues(){
        binding.eq.setText(equation);
        binding.ans.setText(ans);
    }
}
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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">
    <LinearLayout
        android:gravity="end"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/eq"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:layout_marginTop="10dp"
            android:gravity="right|end"
            android:textSize="36sp" />

        <TextView
            android:id="@+id/ans"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:layout_marginTop="10dp"
            android:gravity="right|end"
            android:textSize="36sp" />

        <LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="15dp"
    android:gravity="fill_horizontal|end">

    <Button
        android:id="@+id/clear"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_marginRight="100dp"
        android:backgroundTint="#F76565"
        android:text="C"
        android:textSize="24sp"
        android:textStyle="italic" />

    <Button
        android:id="@+id/delete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:layout_marginLeft="15dp"
        android:background="@drawable/ic_baseline_backspace_24"
        android:textSize="20sp"
        android:textStyle="normal"
        tools:ignore="SpeakableTextPresentCheck" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center">

    <Button
        android:id="@+id/seven"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="7"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/eight"
```

```
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="8"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/nine"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="9"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/divide"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#D015921A"
        android:text="/"
        android:textSize="24sp"
        android:textStyle="bold" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center">
    <Button
        android:id="@+id/four"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="4"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/five"
        android:layout_width="0dp"
```

```
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="5"
        android:textSize="24sp"
        android:textStyle="bold" /
    <Button
        android:id="@+id/six"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="6"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/multiply"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#D015921A"
        android:text="X"
        android:textSize="24sp"
        android:textStyle="bold" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center">
    <Button
        android:id="@+id/one"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="1"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/two"
        android:layout_width="0dp"
```

```
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="2"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/three"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#20988C"
        android:text="3"
        android:textSize="24sp"
        android:textStyle="bold" />
    <Button
        android:id="@+id/sub"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#D015921A"
        android:text="-"
        android:textSize="24sp"
        android:textStyle="bold" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center">
    <Button
        android:id="@+id/dot"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_weight="1"
        android:backgroundTint="#43BEF5"
        android:text="."
        android:textSize="24sp"
        android:textStyle="italic" />

    <Button
        android:id="@+id/zero"
```

```
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_margin="5dp"
            android:layout_weight="1"
            android:backgroundTint="#20988C"
            android:text="0"
            android:textSize="24sp"
            android:textStyle="bold" />

        <Button
            android:id="@+id/equal"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_margin="5dp"
            android:layout_weight="1"
            android:backgroundTint="#43BEF5"
            android:text="="
            android:textSize="24sp"
            android:textStyle="italic" />

        <Button
            android:id="@+id/add"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_margin="5dp"
            android:layout_weight="1"
            android:backgroundTint="#D015921A"
            android:text="+"
            android:textSize="24sp"
            android:textStyle="bold" />
    </LinearLayout>
  </LinearLayout>
</ScrollView>
```

EvaluateString.java

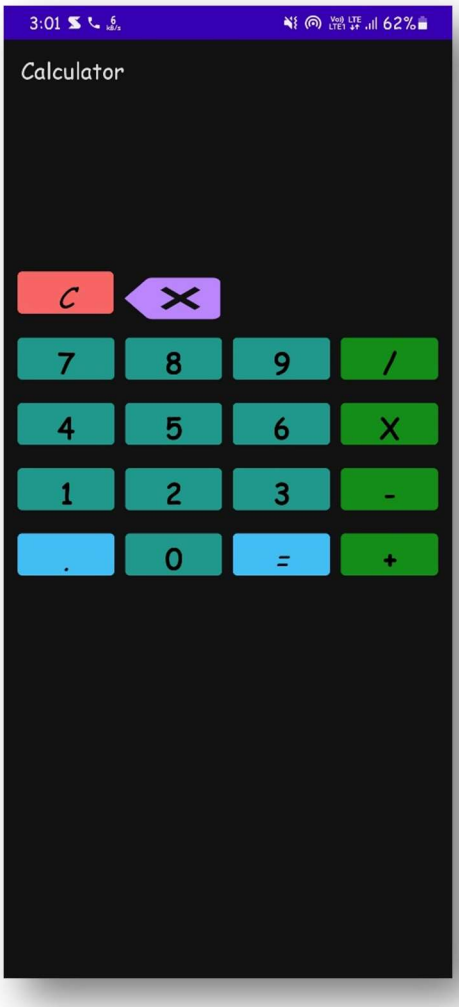
```
package com.paratparinfotech.calculator;
/* A Java program to evaluate a
given expression where tokens
are separated by space.
*/
import java.util.Stack;

public class EvaluateString
{
  public static int evaluate(String expression)
  {
```

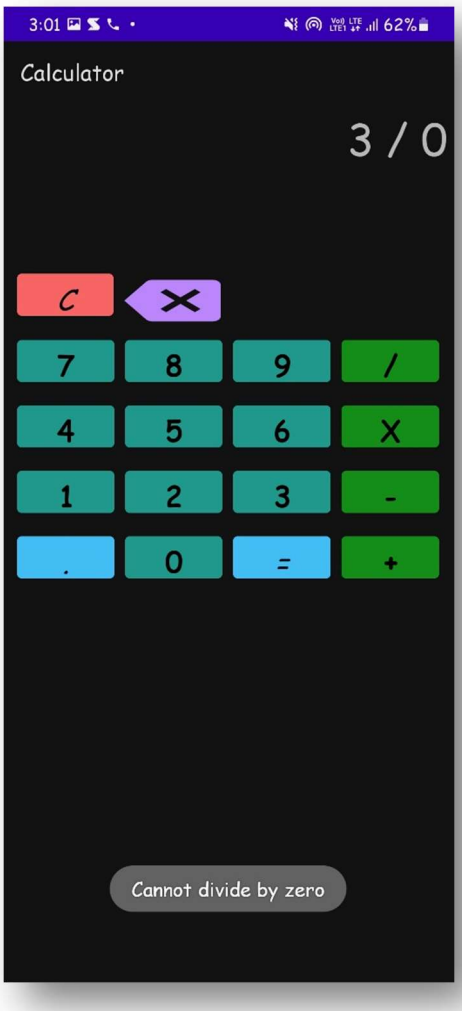
```
char[] tokens = expression.toCharArray();

// Stack for numbers: 'values'
Stack<Integer> values = new
      Stack<Integer>();

// Stack for Operators: 'ops'
Stack<Character> ops = new
      Stack<Character>();

for (int i = 0; i < tokens.length; i++)
{

   // Current token is a
   // whitespace, skip it
   if (tokens[i] == ' ')
      continue;

   // Current token is a number,
   // push it to stack for numbers
   if (tokens[i] >= '0' &&
         tokens[i] <= '9')
   {
      StringBuffer sbuf = new
            StringBuffer();

      // There may be more than one
      // digits in number
      while (i < tokens.length &&
            tokens[i] >= '0' &&
            tokens[i] <= '9')
         sbuf.append(tokens[i++]);
      values.push(Integer.parseInt(sbuf.
            toString()));

      // right now the i points to
      // the character next to the digit,
      // since the for loop also increases
      // the i, we would skip one
      // token position; we need to
      // decrease the value of i by 1 to
      // correct the offset.
      i--;
   }

   // Current token is an opening brace,
   // push it to 'ops'
   else if (tokens[i] == '(')
```

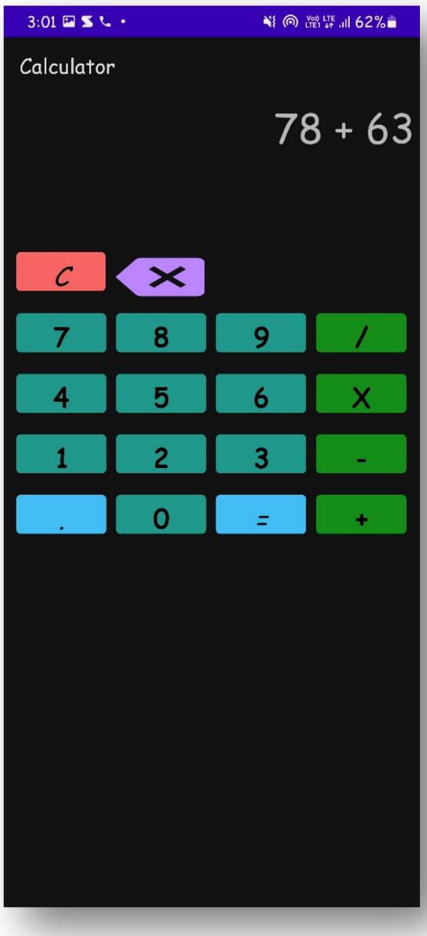
```
            ops.push(tokens[i]);

            // Closing brace encountered,
            // solve entire brace
        else if (tokens[i] == ')')
        {
            while (ops.peek() != '(')
                values.push(applyOp(ops.pop(),
                        values.pop(),
                        values.pop()));
            ops.pop();
        }

        // Current token is an operator.
        else if (tokens[i] == '+' ||
                tokens[i] == '-' ||
                tokens[i] == '*' ||
                tokens[i] == '/')
        {
            // While top of 'ops' has same
            // or greater precedence to current
            // token, which is an operator.
            // Apply operator on top of 'ops'
            // to top two elements in values stack
            while (!ops.empty() &&
                    hasPrecedence(tokens[i],
                            ops.peek()))
                values.push(applyOp(ops.pop(),
                        values.pop(),
                        values.pop()));

            // Push current token to 'ops'.
            ops.push(tokens[i]);
        }
    }

    // Entire expression has been
    // parsed at this point, apply remaining
    // ops to remaining values
    while (!ops.empty())
        values.push(applyOp(ops.pop(),
                values.pop(),
                values.pop()));

    // Top of 'values' contains
    // result, return it
    return values.pop();
}
```

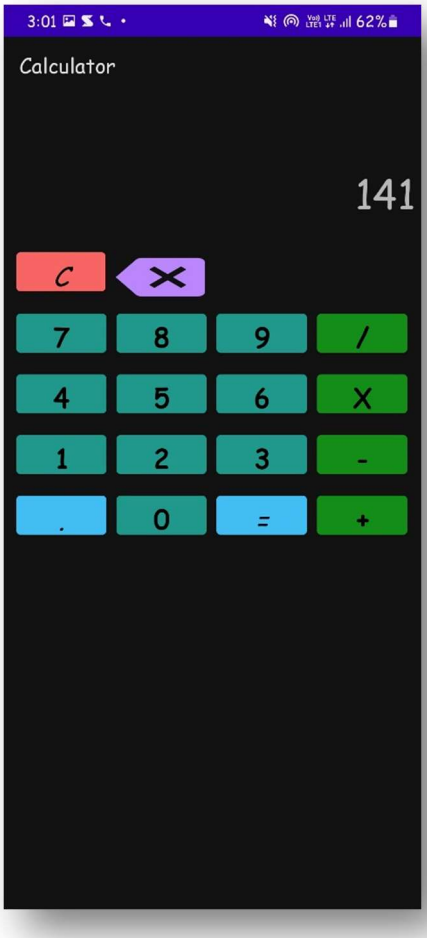
```
    // Returns true if 'op2' has higher
    // or same precedence as 'op1',
    // otherwise returns false.
    public static boolean hasPrecedence(
            char op1, char op2)
    {
        if (op2 == '(' || op2 == ')')
            return false;
        if ((op1 == '*' || op1 == '/') &&
                (op2 == '+' || op2 == '-'))
            return false;
        else
            return true;
    }


    public static int applyOp(char op,
                              int b, int a)
    {
        switch (op)
        {
            case '+':
                return a + b;
            case '-':
                return a - b;
            case '*':
                return a * b;
            case '/':
                if (b == 0)
                    throw new
                        UnsupportedOperationException(
                        "Cannot divide by zero");
                return a / b;
        }
        return 0;
    }
}
```

Output


3:01
62%
Calculator
C
7
8
9
/
4
5
6
X
1
2
3
-
.
0
=
+


3:01
62%
Calculator
3 / 0
C
7
8
9
/
4
5
6
X
1
2
3
-
.
0
=
+
Cannot divide by zero


3:01
62%
Calculator
78 + 63
C
7
8
9
/
4
5
6
X
1
2
3
-
.
0
=
+


3:01
62%
Calculator
141
C
7
8
9
/
4
5
6
X
1
2
3
-
.
0
=
+