**(BCA V SEM)**

**Android Application Development – II**

**Practical Questions**

1. **Write a program in android to create toast.**

**Answer:**

**Activity\_main.xml**

<Button  
 android:id="@+id/btnToast"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Show Toast"  
 tools:layout\_editor\_absoluteX="168dp"  
 tools:layout\_editor\_absoluteY="248dp" />  
  
<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" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {

Button btnToast;

protected void onCreate(Bundle savedInstanceState) {

btnToast = findViewById(R.id.btnToast);  
btnToast.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

Toast.makeText(getApplicationContext(), "This is my First Toast", Toast.LENGTH\_LONG).show();

}  
});

}

}

1. **Write a program in android to open LinkedIn app when the button is clicked.**

**Answer:**

**Activity\_main.xml**

<Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="120dp"  
 android:layout\_marginBottom="332dp"  
 android:text="linkedIn"  
 android:onClick="linkedInbtn"  
 android:textSize="34sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {

public void linkedInbtn(View view){

Uri uri = Uri.parse("https://in.linkedin.com/");

Intent intent = new Intent(Intent.ACTION\_VIEW,uri);

startActivity(intent);  
}

}

1. **Write an Android program that allows selecting time using a TimePicker widget and displays the selected time in a TextView.**

**Answer:**

**activity\_main.xml**

<TimePicker  
 android:id="@+id/timePicker"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/txtTime"  
 app:layout\_constraintTop\_toBottomOf="@+id/timePicker" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {

TimePicker timePicker;  
TextView timeTxt;

timePicker = findViewById(R.id.timePicker);  
timeTxt = findViewById(R.id.txtTime);  
  
timePicker.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {

@Override  
 public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {  
 timeTxt.setText(hourOfDay + " : " + minute);  
 }  
});

}

1. **Write a program in android to create a clickable text that opens the Google.**

**Answer:**

**activity\_main.xml**

<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/txtTime"  
 app:layout\_constraintTop\_toBottomOf="@+id/timePicker"

android:autoLink="all"

android:text="www.google.com" />

1. **Write an Android program that allows selecting date using a material button widget and displays the selected date in a TextView.**

**Answer:**

**Code 1:**

**Activity\_main.xml**

<DatePicker  
 android:id="@+id/datePicker1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginEnd="65dp"  
 android:layout\_marginBottom="283dp"  
 android:calendarViewShown="false"  
 android:datePickerMode="spinner" />  
  
<Button  
 android:text="Submit"  
 android:id="@+id/btnSubmit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginEnd="135dp"  
 android:layout\_marginBottom="175dp" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {

DatePicker datePicker;  
Button button1;

datePicker = findViewById(R.id.datePicker1);  
button1 = findViewById(R.id.btnSubmit);  
  
button1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String day, month, year;  
 day = String.valueOf(datePicker.getDayOfMonth());  
 month = String.valueOf(datePicker.getMonth() + 1);  
 year = String.valueOf(datePicker.getYear());  
 Toast.makeText(MainActivity.this, day + "/" + month + "/" + year, Toast.LENGTH\_SHORT).show();  
 }  
});

**Code 2:**

**Activity\_main.xml**

<com.google.android.material.button.MaterialButton  
 android:id="@+id/datePicker"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="164dp"  
 android:layout\_marginTop="239dp"  
 android:text="Date Picker" />  
  
<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginEnd="188dp"  
 android:layout\_marginBottom="131dp"  
 android:id="@+id/tv" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {

MaterialButton button;

TextView textView;

protected void onCreate(Bundle savedInstanceState) {

button = findViewById(R.id.datePicker);

textView = findViewById(R.id.tv);

button.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

MaterialDatePicker<Long> materialDatePicker = MaterialDatePicker.Builder.datePicker()  
.setTitleText("Select Date")

.setSelection(MaterialDatePicker

.todayInUtcMilliseconds()).build();

materialDatePicker.addOnPositiveButtonClickListener(new MaterialPickerOnPositiveButtonClickListener<Long>() {  
  
public void onPositiveButtonClick(Long selection) {

String date = new SimpleDateFormat("MM-dd-yyy",

Locale.getDefault()).format(new Date(selection));

textView.setText(MessageFormat.format("Selected Date:{0}",

date));

}

});  
 materialDatePicker.show(getSupportFragmentManager(),"tag");  
 }  
 });  
 }  
}

1. **Write a program to send one activity data to another activity.**

**activity\_main.xml**

<EditText  
 android:id="@+id/name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:ems="10"  
 android:hint="Name"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.437"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.069" />  
  
<Button  
 android:id="@+id/ADD"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="145dp"  
 android:layout\_marginTop="144dp"  
 android:text="Add"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/name" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 private EditText name;  
 private Button add;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 name = findViewById(R.id.name);  
 add = findViewById(R.id.ADD);  
  
 add.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String username = name.getText().toString();  
  
 Intent intent = new Intent(MainActivity.this,MainActivity2.class);  
 intent.putExtra("keyname",username);  
 startActivity(intent);  
 }  
 });  
 }  
}

**activity\_main2.xml**

<TextView  
 android:id="@+id/text\_name"  
 android:layout\_width="161dp"  
 android:layout\_height="35dp"  
 android:layout\_marginTop="120dp"  
 android:text="Name"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

**MainActivity2.java**

public class MainActivity2 extends AppCompatActivity {  
  
 private TextView name;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main2);  
  
 name = findViewById(R.id.text\_name);  
  
 String username = getIntent().getStringExtra("keyname");  
 name.setText(username);  
 }  
}

1. **Write a program to create alert dialog.**

**Activity\_main.xml**

<Button  
 android:id="@+id/my\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="140dp"  
 android:layout\_marginTop="256dp"  
 android:text="Hello World!"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 Button myButton;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 myButton = findViewById(R.id.my\_button);

myButton.setOnClickListener(new View.OnClickListener() {

@Override  
 public void onClick(View view) {

AlertDialog dialog = createDialog();  
 dialog.show();  
  
 }  
 });  
 }  
 AlertDialog createDialog(){

AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setMessage("Do you want to do?");

builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override  
 public void onClick(DialogInterface dialogInterface, int i) {

Toast.makeText(MainActivity.this,"Clicked Yes",Toast.LENGTH\_LONG).show();  
 }  
 });  
 builder.setNegativeButton("No Please", new DialogInterface.OnClickListener() {

@Override  
 public void onClick(DialogInterface dialogInterface, int i) {

Toast.makeText(MainActivity.this,"Clicked No",Toast.LENGTH\_LONG).show();  
  
 }  
 });  
 return builder.create();  
 }  
}

1. **Write a program to create Touch Mode Application.**

**Code 1:**

**Activity\_main.xml**

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 ConstraintLayout myLayout = (ConstraintLayout) findViewById(R.id.main\_layout);  
 myLayout.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View v, MotionEvent event) {  
 Log.i("TouchEvents","Touch is Detected");  
 return true;  
 }  
 });  
 }  
}

**Code 2:**

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 ConstraintLayout myLayout = (ConstraintLayout) findViewById(R.id.main\_layout);  
 myLayout.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View v, MotionEvent event) {  
  
 int eventType = event.getActionMasked();  
 switch(eventType){  
 case MotionEvent.ACTION\_DOWN:  
 Log.i("TouchEvents","Action Down");  
 break;  
  
 case MotionEvent.ACTION\_UP:  
 Log.i("TouchEvents","Action Up");  
 break;  
  
 case MotionEvent.ACTION\_MOVE:  
 Log.i("TouchEvents","Action Move");  
 break;  
 }  
 return true;  
 }  
 });  
 }  
}

**Code 3:**

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 ConstraintLayout myLayout = (ConstraintLayout) findViewById(R.id.main\_layout);  
 myLayout.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View v, MotionEvent event) {  
  
 int eventType = event.getActionMasked();  
 switch(eventType){  
 case MotionEvent.ACTION\_DOWN:  
 Log.i("TouchEvents","Action Down");  
 break;  
  
 case MotionEvent.ACTION\_POINTER\_DOWN:  
 Log.i("TouchEvents","Action Pointer Down " + event.getPointerCount());  
 break;  
  
 case MotionEvent.ACTION\_UP:  
 Log.i("TouchEvents","Action Up");  
 break;  
  
 case MotionEvent.ACTION\_POINTER\_UP:  
 Log.i("TouchEvents","Action Pointer UP " + event.getPointerCount());  
 break;  
  
 case MotionEvent.ACTION\_MOVE:  
 Log.i("TouchEvents","Action Move");  
 break;  
 }  
 return true;  
 }  
 });  
 }  
}

1. **Write a program to set Image on ImageView**

**Activity\_main.xml**

<ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="284dp"  
 android:layout\_height="256dp"  
 android:layout\_marginStart="72dp"  
 android:layout\_marginTop="68dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:srcCompat="@tools:sample/avatars" />  
  
<Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="156dp"  
 android:layout\_marginTop="436dp"  
 android:text="Click To Set"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 Button button = (Button) findViewById(R.id.button);  
 button.setOnClickListener((v) ->{  
 ImageView imgview = (ImageView) findViewById(R.id.imageView);  
 imgview.setBackgroundResource(R.drawable.photo1);  
  
 });  
 }  
}

1. **Write a program to take pictures from camera and set the image on image View.**

**Activity\_main.xml**

<TextView  
 android:id="@+id/text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="40dp"  
 android:layout\_marginTop="71dp"  
 android:text="Click on the camera Button to start the camera"  
 android:textSize="16sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
<Button  
 android:id="@+id/btncamera\_id"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/text"  
 android:layout\_marginStart="156dp"  
 android:layout\_marginTop="140dp"  
 android:text="Camera"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
<ImageView  
 android:id="@+id/imageview1"  
 android:layout\_width="231dp"  
 android:layout\_height="329dp"  
 android:layout\_below="@+id/btncamera\_id"  
 android:layout\_marginStart="90dp"  
 android:layout\_marginTop="60dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btncamera\_id" />

**MainActivity.java**

public class MainActivity extends AppCompatActivity {  
  
 private static final int REQUEST\_CODE = 22;  
 Button btnpicture;  
 ImageView imageview;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 btnpicture = findViewById(R.id.btncamera\_id);  
 imageview = findViewById(R.id.imageview1);  
  
 btnpicture.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent cameraIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  
 startActivityForResult(cameraIntent,REQUEST\_CODE);  
 }  
 });  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 if (requestCode == REQUEST\_CODE && resultCode == RESULT\_OK){  
 Bitmap photo = (Bitmap) data.getExtras().get("data");  
 imageview.setImageBitmap(photo);  
 } else {  
 Toast.makeText(this,"Cancelled", Toast.LENGTH\_SHORT).show();  
 super.onActivityResult(requestCode, resultCode, data);  
 }  
 }  
}

**AndroidManifest.xml**

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

1. **Write a program to on and off Bluetooth, show paired devices and discover devices**

**Activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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"  
 android:orientation="vertical"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 tools:context="com.mcuhq.simplebluetooth.MainActivity">  
  
 <CheckBox  
 android:id="@+id/checkbox\_led\_1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginTop="10dp"  
 android:checked="false"  
 android:scaleX="1.7"  
 android:scaleY="1.7"  
 android:text="Toggle LED"  
 android:textStyle="bold" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.1"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:text="RX:"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/read\_buffer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.9"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:text="&lt;Read Buffer>" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.1"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:text="Status:"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/bluetooth\_status"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.9"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:text="&lt;Bluetooth Status>" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/scan"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Bluetooth ON" />  
  
 <Button  
 android:id="@+id/off"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Bluetooth OFF" />  
  
 <Button  
 android:id="@+id/paired\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Show paired Devices" />  
  
 <Button  
 android:id="@+id/discover"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Discover New Devices" />  
  
 <ListView  
 android:id="@+id/devices\_list\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:choiceMode="singleChoice" />  
</LinearLayout>

**Main\_Activity.java**

package com.mcuhq.simplebluetooth;  
  
import android.Manifest;  
import android.bluetooth.BluetoothAdapter;  
import android.bluetooth.BluetoothDevice;  
import android.bluetooth.BluetoothSocket;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Looper;  
import android.os.Message;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.content.ContextCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.CheckBox;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.io.IOException;  
import java.io.UnsupportedEncodingException;  
import java.lang.reflect.Method;  
import java.nio.charset.StandardCharsets;  
import java.util.Set;  
import java.util.UUID;  
  
public class MainActivity extends AppCompatActivity {  
  
 private final String TAG = MainActivity.class.getSimpleName();  
  
 private static final UUID BT\_MODULE\_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB"); // "random" unique identifier  
  
 // #defines for identifying shared types between calling functions  
 private final static int REQUEST\_ENABLE\_BT = 1; // used to identify adding bluetooth names  
 public final static int MESSAGE\_READ = 2; // used in bluetooth handler to identify message update  
 private final static int CONNECTING\_STATUS = 3; // used in bluetooth handler to identify message status  
  
 // GUI Components  
 private TextView mBluetoothStatus;  
 private TextView mReadBuffer;  
 private Button mScanBtn;  
 private Button mOffBtn;  
 private Button mListPairedDevicesBtn;  
 private Button mDiscoverBtn;  
 private ListView mDevicesListView;  
 private CheckBox mLED1;  
  
 private BluetoothAdapter mBTAdapter;  
 private Set<BluetoothDevice> mPairedDevices;  
 private ArrayAdapter<String> mBTArrayAdapter;  
  
 private Handler mHandler; // Our main handler that will receive callback notifications  
 private ConnectedThread mConnectedThread; // bluetooth background worker thread to send and receive data  
 private BluetoothSocket mBTSocket = null; // bi-directional client-to-client data path  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 mBluetoothStatus = (TextView)findViewById(R.id.bluetooth\_status);  
 mReadBuffer = (TextView) findViewById(R.id.read\_buffer);  
 mScanBtn = (Button)findViewById(R.id.scan);  
 mOffBtn = (Button)findViewById(R.id.off);  
 mDiscoverBtn = (Button)findViewById(R.id.discover);  
 mListPairedDevicesBtn = (Button)findViewById(R.id.paired\_btn);  
 mLED1 = (CheckBox)findViewById(R.id.checkbox\_led\_1);  
  
 mBTArrayAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1);  
 mBTAdapter = BluetoothAdapter.getDefaultAdapter(); // get a handle on the bluetooth radio  
  
 mDevicesListView = (ListView)findViewById(R.id.devices\_list\_view);  
 mDevicesListView.setAdapter(mBTArrayAdapter); // assign model to view  
 mDevicesListView.setOnItemClickListener(mDeviceClickListener);  
  
 // Ask for location permission if not already allowed  
 if(ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED)  
 ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS\_COARSE\_LOCATION}, 1);  
  
  
 mHandler = new Handler(Looper.getMainLooper()){  
 @Override  
 public void handleMessage(Message msg){  
 if(msg.what == MESSAGE\_READ){  
 String readMessage = null;  
 readMessage = new String((byte[]) msg.obj, StandardCharsets.UTF\_8);  
 mReadBuffer.setText(readMessage);  
 }  
  
 if(msg.what == CONNECTING\_STATUS){  
 char[] sConnected;  
 if(msg.arg1 == 1)  
 mBluetoothStatus.setText(getString(R.string.BTConnected) + msg.obj);  
 else  
 mBluetoothStatus.setText(getString(R.string.BTconnFail));  
 }  
 }  
 };  
  
 if (mBTArrayAdapter == null) {  
 // Device does not support Bluetooth  
 mBluetoothStatus.setText(getString(R.string.sBTstaNF));  
 Toast.makeText(getApplicationContext(),getString(R.string.sBTdevNF),Toast.LENGTH\_SHORT).show();  
 }  
 else {  
  
 mLED1.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 if(mConnectedThread != null) //First check to make sure thread created  
 mConnectedThread.write("1");  
 }  
 });  
  
  
 mScanBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 bluetoothOn();  
 }  
 });  
  
 mOffBtn.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 bluetoothOff();  
 }  
 });  
  
 mListPairedDevicesBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v){  
 listPairedDevices();  
 }  
 });  
  
 mDiscoverBtn.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 discover();  
 }  
 });  
 }  
 }  
  
 private void bluetoothOn(){  
 if (!mBTAdapter.isEnabled()) {  
 Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);  
 startActivityForResult(enableBtIntent, REQUEST\_ENABLE\_BT);  
 mBluetoothStatus.setText(getString(R.string.BTEnable));  
 Toast.makeText(getApplicationContext(),getString(R.string.sBTturON),Toast.LENGTH\_SHORT).show();  
  
 }  
 else{  
 Toast.makeText(getApplicationContext(),getString(R.string.BTisON), Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 // Enter here after user selects "yes" or "no" to enabling radio  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent Data){  
 // Check which request we're responding to  
 if (requestCode == REQUEST\_ENABLE\_BT) {  
 // Make sure the request was successful  
 if (resultCode == RESULT\_OK) {  
 // The user picked a contact.  
 // The Intent's data Uri identifies which contact was selected.  
 mBluetoothStatus.setText(getString(R.string.sEnabled));  
 }  
 else  
 mBluetoothStatus.setText(getString(R.string.sDisabled));  
 }  
 }  
  
 private void bluetoothOff(){  
 mBTAdapter.disable(); // turn off  
 mBluetoothStatus.setText(getString(R.string.sBTdisabl));  
 Toast.makeText(getApplicationContext(),"Bluetooth turned Off", Toast.LENGTH\_SHORT).show();  
 }  
  
 private void discover(){  
 // Check if the device is already discovering  
 if(mBTAdapter.isDiscovering()){  
 mBTAdapter.cancelDiscovery();  
 Toast.makeText(getApplicationContext(),getString(R.string.DisStop),Toast.LENGTH\_SHORT).show();  
 }  
 else{  
 if(mBTAdapter.isEnabled()) {  
 mBTArrayAdapter.clear(); // clear items  
 mBTAdapter.startDiscovery();  
 Toast.makeText(getApplicationContext(), getString(R.string.DisStart), Toast.LENGTH\_SHORT).show();  
 registerReceiver(blReceiver, new IntentFilter(BluetoothDevice.ACTION\_FOUND));  
 }  
 else{  
 Toast.makeText(getApplicationContext(), getString(R.string.BTnotOn), Toast.LENGTH\_SHORT).show();  
 }  
 }  
 }  
  
 final BroadcastReceiver blReceiver = new BroadcastReceiver() {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 if(BluetoothDevice.ACTION\_FOUND.equals(action)){  
 BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA\_DEVICE);  
 // add the name to the list  
 mBTArrayAdapter.add(device.getName() + "\n" + device.getAddress());  
 mBTArrayAdapter.notifyDataSetChanged();  
 }  
 }  
 };  
  
 private void listPairedDevices(){  
 mBTArrayAdapter.clear();  
 mPairedDevices = mBTAdapter.getBondedDevices();  
 if(mBTAdapter.isEnabled()) {  
 // put it's one to the adapter  
 for (BluetoothDevice device : mPairedDevices)  
 mBTArrayAdapter.add(device.getName() + "\n" + device.getAddress());  
  
 Toast.makeText(getApplicationContext(), getString(R.string.show\_paired\_devices), Toast.LENGTH\_SHORT).show();  
 }  
 else  
 Toast.makeText(getApplicationContext(), getString(R.string.BTnotOn), Toast.LENGTH\_SHORT).show();  
 }  
  
 private AdapterView.OnItemClickListener mDeviceClickListener = new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
  
 if(!mBTAdapter.isEnabled()) {  
 Toast.makeText(getBaseContext(), getString(R.string.BTnotOn), Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 mBluetoothStatus.setText(getString(R.string.cConnet));  
 // Get the device MAC address, which is the last 17 chars in the View  
 String info = ((TextView) view).getText().toString();  
 final String address = info.substring(info.length() - 17);  
 final String name = info.substring(0,info.length() - 17);  
  
 // Spawn a new thread to avoid blocking the GUI one  
 new Thread()  
 {  
 @Override  
 public void run() {  
 boolean fail = false;  
  
 BluetoothDevice device = mBTAdapter.getRemoteDevice(address);  
  
 try {  
 mBTSocket = createBluetoothSocket(device);  
 } catch (IOException e) {  
 fail = true;  
 Toast.makeText(getBaseContext(), getString(R.string.ErrSockCrea), Toast.LENGTH\_SHORT).show();  
 }  
 // Establish the Bluetooth socket connection.  
 try {  
 mBTSocket.connect();  
 } catch (IOException e) {  
 try {  
 fail = true;  
 mBTSocket.close();  
 mHandler.obtainMessage(CONNECTING\_STATUS, -1, -1)  
 .sendToTarget();  
 } catch (IOException e2) {  
 //insert code to deal with this  
 Toast.makeText(getBaseContext(), getString(R.string.ErrSockCrea), Toast.LENGTH\_SHORT).show();  
 }  
 }  
 if(!fail) {  
 mConnectedThread = new ConnectedThread(mBTSocket, mHandler);  
 mConnectedThread.start();  
  
 mHandler.obtainMessage(CONNECTING\_STATUS, 1, -1, name)  
 .sendToTarget();  
 }  
 }  
 }.start();  
 }  
 };  
  
 private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException {  
 try {  
 final Method m = device.getClass().getMethod("createInsecureRfcommSocketToServiceRecord", UUID.class);  
 return (BluetoothSocket) m.invoke(device, BT\_MODULE\_UUID);  
 } catch (Exception e) {  
 Log.e(TAG, "Could not create Insecure RFComm Connection",e);  
 }  
 return device.createRfcommSocketToServiceRecord(BT\_MODULE\_UUID);  
 }  
}

**AndroidManifests.xml**

<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
  
<permission android:name="android.permission.BLUETOOTH" android:label="BLUETOOTH" />  
<permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
<permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />