5-3-2021

① Create an application to display "Hello World"

MainActivity.java

```
package com.example.myapp;
import androidx.appcompat.app.AppCompat
                                    Activity;
import android.content.Intent;
import android.view.View;
public class MainActivity extends AppCompat
                                    Activity{

    @Override
    protected void onCreate(Bundle
                            savedInstance
                            State)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_
                            main);

    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android x.constraintlayout.widget.ConstraintLayout
    xmlns:"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">
```

```
<TextView
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "Helloworld"
    android: id = "@/heading"
    app:layout_constraintBottom_toBottomOf = "parent
    app:layout_constraintLeft_toLeftOf = "parent
    app:layout_constraintTop_toTopOf = "parent />

</androidx.constraintlayout.widget.ConstraintLayout>
```

12-3-21

(2) Illustrate life cycle of an application with an app.

(2)

```
package com.example.myapp2;
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(tag: "Lifecycle", msg: "onCreate invoked");
    }
    @Override
    protected void onStart(){
        super.onStart();
        Log.d(tag: "lifecycle2", msg: "onStart invoked");
    }
    @Override
    protected void onResume(){
        super.onResume();
        Log.d("Lifecycle3", "onResume invoked");
    }
    @Override
    protected void onPause(){
        super.onPause();
        Log.d("Life cycle 4", msg: "On pause invoked");
    }
}
```

```
@Override
protected void onStop(){
    super.onStop();
    Log.d("lifecycle 5","onStop invoked");
}
@Override
protected void onRestart(){
        super.onRestart();
        Log.d("lifecycle 6","onRestart invoked");
}
@Override
protected void onDestroy(){
        super.onDestroy();
        Log.d("lifecycle 7","onDestroy invoked");
}
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintLayout.widget.ConstraintLayout
    xmlns:app="http://schemas.android.com/apk/res/android"
    :
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@id/heading"
        android:text="Application lifecycle"
        android:layout_constraintTop_toTopOf="parent"
        android:layout_constraintLeft_toLeftOf="parent"
        android:layout_constraintRight_toRightOf="parent" />

</androidx.constraintLayout.widget.ConstraintLayout>
```

O/P: In Logcat

on opening.

On Create invoked

OnStart invoked

On Resume invoked

On Pause invoked

On Stop invoked

On Destroy invoked

3) To understand Activity & intent create a sample application with login module. (Check username & password), on successful login, go to next screen & on failing login & alert the user using toast.
Also pass username to next screen (Intents)

MainActivity.java

```
package comexample.intent.
import androidx.appcompat.app.actionBar
import androidx.appcompat.app.*;
import android.content.*;
import android.os.*;
import android.view.View;
import android.widget.*;

public class MainActivity extends AppCompatActivity{
    @Override
    protected void onCreate(Bundle saved){
        super.onCreate(saved);
        setContentView(R.layout.activity_main);
```

```
ActionBar a = getSupportActionBar();
a.setTitle("Main Activity");
EditText uemail = findViewById(R.id.email);
EditText ps = findViewById(R.id.password);
Button btn = findViewById(R.id.button);

btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v){
        String email = uemail.getText().toString();
        String password = ps.getText().toString();
        if(email.equals("neeraj.badam@gmail.com") &&
                password.equals("bneeraj27")){
            Toast.makeText(MainActivity.this,
                    "Successfully Logged in", Toast.LENGTH_
                    SHORT).show();
            Intent intent = new Intent(MainActivity.this,
                    SecondActivity.class);
            intent.putExtra("Email", email);
            intent.putExtra("Password", password);
            startActivity(intent);
        }
        else{
            Toast.makeText(MainActivity.this, "Invalid
            E-mail/Username/Password", Toast.LENGTH_SHORT).
```

```
        show();
    }
    }
});
}
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="...../android"
    xmlns:app=.....
    xmlns:tools=.....
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    :
    android:padding="20dp"
    >
    <EditText
        android:id="@+id/email"
        android:inputType="textEmailAddress"
        :
    />
    <EditText
        android:id="@+id/password"
        android:inputType="textPassword"
        :
```

```
<Button
    android:id ="@id/button"
    android:text ="Login"/>
    :
/>
</LinearLayout>
```

SecondActivity.java

```
package com.example.intents;
import ...
public class SecondActivity extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle instance){
        super.onCreate(instance);
        setContentView(R.layout.activity_second);
        ActionBar a = getSupportActionBar();
        a.setTitle("Second Activity");
        Intent i = getIntent();
        String email = i.getStringExtra("Email");
        String pass = i.getStringExtra("Password");
        TextView text = findViewById(R.id.display);
        text.setText("E-mail : "+email+"\nPassword : "+
                                                pass);
```

activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout ...
    :
    :
    android:orientation="vertical">
    android:padding="10dp">

    <TextView
        android:id="@id/display"
        :
        android:text="TextView" />

</LinearLayout>
```

4) Create login app, where you will have to validate email Id (username. Till username & password is not validated, login button should be disabled. On successful login, open browser URL. And on failing login, alert user using Toast (LoginApp).

MainActivity.java

```
package com.example.loginapp;
import ...

public class MainActivity extends Activity OnClickListener,
                                    TextWatcher {
    EditText name;
    EditText pass;
    Button login;
    Button cancel;
    @Override
    protected void onCreate(Bundle instance){
        super.onCreate(instance);
        setContentView (R.layout.activity_main);
        name =(EditText) findViewById(R.id.editText1);
        name.addTextChangedListener(this);
```

```
pass = (EditText) findViewById(R.id.edittext2);
pass.addTextChangedListener(this);
login = (Button) findViewById(R.id.button1);
cancel = (Button) findViewById(R.id.button2);
login.setOnClickListener(this);
cancel.setOnClickListener(this);
login.setEnabled(false);
cancel.setEnabled(true);
}
public void afterTextChanged(Editable arg0){
}
public void beforeTextChanged(CharSequence arg0,
int arg1, int arg2, int arg3){
}
public void onTextChanged(CharSequence arg0, int arg1,
int arg1, int arg2, int arg3){
String na = name.getText().toString();
String pa = pass.getText().toString();
if(na.equals("teeraj") && pa.equals("root") ||
pa.equals("admin")){
Button btn = (Button) findViewById(R.id.
button1);
btn.setEnabled(true);
}
}
```

```
@SuppressLint({"ShowToast", "InlinedApi"})
public void onClick(View v){
    String na = name.getText().toString();
    String pa = pass.getText().toString();
    switch(v.getId())
    {
        case R.id.button1:
            if(na.equals("neeraj")&&
              pa.equals("root")||pa.equals("admin
                                          ")){
            Intent b = new Intent(Intent.
                                  ACTION_VIEW
            , Uri.parse("http://www.google.com"));
            startActivity(b);
        }
        else{
          Toast.makeText("Invalid Username/
                                    password",
                                  3600).show();
        }
        break;

    case R.id.button2: name.setText("");
                       pass.setText("");
                       break;
    default:           break;
    }
}
```

activity_main.xml

```
<RelativeLayout ...
        :
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView1"
        :
    "?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/editText1"
        :
        android:ems="10"
        android:inputType="text" />
    <TextView
        android:id="@+id/textView2"

        :
        />
    <EditText
        android:id="@+id/editText2"
        :
        android:inputType="textPassword" />

    <Button
        android:id="@+id/button1"
        :
        />
    <Button
        @android:id="@+id/button2"
        :
        />
</RelativeLayout>
```

5) Create an application (Using Implicit intents)

- To call specific entered number by user in the EditText
- To open any URL inside the application

(Implicit Intents)

activity_main.xml

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

<LinearLayout ...
    :
    :
    android:orientation="vertical" >

    <EditText
        android:id="@id/editTextPhone"
        android:inputType="phone"
        :
        />

    <Button
        android:id="@id/button"
        android:text="Make a call"
        :
        />

    <Button
        android:id="@id/btn_load"
        android:text="Open Google"
    />

</LinearLayout>
```

add

<uses-permission android:name="android.permission.CALL_PHONE" />

to androidManifest file

& after running app set permissions for making call.

MainActivity.java

```
package com.example.implicitintents;

import ...
public class MainActivity extends AppCompatActivity{

    @Override
    provide void onCreate(Bundle instance) {

        super.onCreate(instance);
        setContentView(R.layout.activity_main);
        EditText ph = findViewById(R.id.editTextPhone);
        Button btn = findViewById(R.id.button);

    btn.setOnClickListener(new View.
                    onClickListener(){

        @Override
        public void onClick(View v){
```

```
String n = ph.getText().toString();
if(n.length() == 10){
    Intent i = new Intent(Intent.
                          ACTION_CALL);
    i.setData(Uri.parse("tel:"+n));
    startActivity(i);
}
else{
    Toast.makeText(getApplicationContext(),
        "Please enter correct number",
        Toast.LENGTH_SHORT).show();
}
}
});
Button btn2 = findViewById(R.id.btn_load);
btn2.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        String url="www.google.com";
        Intent i = new Intent(Intent.
                              ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
});}
```

13

1) Create an application to display notifications on status bar

MainActivity.java

```
package com.example.notificationsdemo
import ...
public class MainActivity extends AppCompatActivity{
    Button btn;
    @Override
    protected void onCreate(Bundle instance){
        super.onCreate(instance);
        setContentView(R.layout.activity_main);
        btn = findViewById(R.id.button);
        if(Build.VERSION.SDK_INT >=
                Build.VERSION_CODES.O){
        NotificationChannel channel = new
            NotificationChannel("My Notification", "hello",
                NotificationManager.IMPORTANCE_
                    DEFAULT);
            NotificationManager m = getSystemService
                (NotificationManager.class);
            m.createNotificationChannel(channel);
    }
```

```
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v){
        String m = "Notifications Example";
        NotificationCompat.Builder builder =
        new NotificationCompat.Builder(MainActivity.
        this, "My Notification");
        buider = setContentTitle("My Title");
        builder.setContentText(m);
        builder.setSmallIcon(R.drawable.ic_message)
        builder.setAutoCancel(true);
        NotificationManagerCompat m =
            NotificationManagerCompat.from
                (MainActivity.this);
            m.notify(1, builder.build());
    }
});
}
}
```

main-activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout ...
    :
    :
    />
    <Button
        android:id="@+id/button"
        :
        :
        android:text="Notify" />
</RelativeLayout>
```

(8) Create an application to add fragments to an activity during runtime.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<linearLayout xmlns:---
    ---width---
    --- height---
    android:orientation="vertical">
    <Button
        android:id="@+id/btn_load"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Load fragment"
    />
    <LinearLayout
        android:id="@+id/fragment_container"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        >
    </LinearLayout>
</linearLayout>
```

MainActivity.java

```
package com.example.mad_app4;

import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class MainActivity extends Activity{
    /** Called when the activity is first created.*/
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnLoad = (Button) findViewById
                        (R.id.btn_load);
        onClickListener listener = new OnClick
                                   Listener(){
            @Override
            pubic void onClick(View v){
FragmentManager f= getFragmentManager();
FragmentTransaction ft = f.beginTransaction();
HelloFragment h=new HelloFragment();
ft.add(R.id.fragment_container, hello, "HELLO");
ft.commit();
```

```
        };
        btnLoad.setOnClickListener(listener);
    }
}
```

hello_fragment_layout.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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="This is a fragment"
        android:textColor="#FFF000"
        android:textSize="30dp"
        android:background="#ffF000" />

    <Space
        android:layout_width="match_parent"
        android:layout_height="10dp" />

</LinearLayout>
```

HelloFragment.java

```
package com.example.mad_app4;

import android.app.Fragment;
import android.os.*;
import android.view.*;

pubic class HelloFragment extends Fragment{
    @Override
    public View onCreateView(LayoutInflater
            inflater, ViewGroup container,
            Bundle savedInstanceState){
        View v=inflater.inflate(R.layout.
            hello_fragment_layout, null);

        return v;
    }
}
```

12) Create an application to display a set of images using Gallery view & then display the selected image in image view.

MainActivity.java

```
package com.example.galleryview;

import ...
public class MainActivity extends AppCompatActivity {

    Gallery g;
    CustomizedGalleryAdapter c;
    ImageView s;
    int[] img = { R.drawable.b, R.drawable.c,
                  R.drawable.d, R.drawable.e,
                  R.drawable.f, R.drawable.g };
    @Override
    protected void onCreate(Bundle instance){
        super.onCreate(instance);
        setContentView(R.layout.activity_main);
        g = (Gallery)findViewById(R.id.languages
                                  Gallery);
        s = (ImageView)findViewById(R.id.imageView
        g.setAdapter(c);
        c = new CustomizedGalleryAdapter(getApplicat
                                  -i,onContent(),img
        g.setAdapter(c);
```

```
g.setOnItemClickListener(new AdapterView.On
                        ItemClickListener(){

    @Override
    public void onItemClick(AdapterView<?>
      parent, View v, int pos, long id){
        s.setImageResource(img[pos]);
    }
  });
 }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
 <LinearLayout
      :
      tools:context = ".MainActivity">
     <ImageView
        android:id="@+id/imageView"
        />
     <Gallery
        android:id="@+id/languageGallery"
          :
        />
</LinearLayout>
```

CustomizedGalleryAdapter.java

```
package com.example.galleryview;
import ...
public class CustomizedGalleryAdapter extends BaseAdapter{
    private Context c;
    private int[] i;
    public CustomizedGalleryAdapter(Context c1, int[] i1){
        c = c1;
        i = i1;
    }
    public int getCount(){
        return i.length;
    }
    public Object getItem(int pos){
        return pos;
    }
    public long getItemId(int pos){
        return pos;
    }
    public View getView(int posView c, ViewGroup p){
        ImageView iV = new ImageView(c);
        iV.setImageResource(i[pos]);
        iV.setLayoutParams(new Gallery.LayoutParams
                                (200, 200));
        return iV;
    }
}
```

13) Create an application to display a set of images using Grid View & then display the selected image in another screen.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
        :
        tools:context=".MainActivity">
        <GridView
            android:id="@+id/gridView"
            android:numColumns="2"
            :
            />
</RelativeLayout>
```

row-data.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Relativelayout :
        android:layout_height="match_parent">
        <RelativeLayout
            android:id="@+id/gridViewdata"
            :
            />
            <ImageView
                android:id="@+id/images"
                :
                android:src="@drawable/a" />
        </Relativelayout>
</RelativeLayout>
```

activity_grid_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    tools:context=".GridItemsActivity">

    <ImageView
        android:id="@+id/imageView"
        android:src="@drawable/a" />
</RelativeLayout>
```

GridItemActivity.java

```
package com.example.gridView;
import ...
public class GridItemActivity extends AppCompatActivity{
    ImageView img;
    @Override
    protected void onCreate(Bundle instance){
        super.onCreate(instance);
        setContentView(R.layout.activity_grid_item);
        img = findViewById(R.id.imageView);
        img.setImageResource(intent.getIntExtra("image",
                                                0));
    }
}
```

MainActivity.java

```
package com.example.gridView;
import ...
public class MainActivity extends AppCompatActivity {
    GridView g;
    int img[] = {R.drawable.a, R.drawable.b, ....};
    @Override
    protected void onCreate(Bundle instance){
        super.onCreate(instance);
        setContentView(R.id.activity_main);
    g = findViewById(R.id.gridView);
    CustomAdapter c = new CustomAdapter();
    g.setAdapter(c);
    g.setOnItemClickListener(new AdapterView.On
                                ItemClickListener(){
        @Override
        public void onItemClick(AdapterView<?> p,
                View v, int i, long id){
            Intent i = new Intent(getAdapterContext(),
                                GridItemActivity.class);
            i.putExtra("image", img[i]);
            startActivity(i);
        }
    });
}
private class CustomAdapter extends BaseAdapter {
    @Override
    public int getCount(){
        return img.length;
    }
```

```
@Override
public Object getItem(int p){
    return null;
}
@Override
public long getItemId(int p){
    return 0;
}
@Override
public View getView(int i, View c, ViewGroup p){
    View v = getLayoutInflater().inflate(
                R.layout.row_data, null);
    ImageView image = v.findViewById(R.id.image);
    image.setImageResource(img[i]);
    return v;
  }
}
}
```

14) Create an application that will change color of screen, based on selected options from Options Menu.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
        tools:context=".MainActivity" />
```

OR

main_menu1.xml ⇒ We should create a menu.

```
<?xml version="1.0" encoding="utf-8" ?>
<menu1 xmlns:....
    <item android:id="@id/red"
        app:showAsAction="never" />
    <item android:id="@id/green"
        app:showAsAction="never" />
    <item android:id="@id/blue
        app:showAsAction="never" />
    <item android:id="@id/gray"
        app:showAsAction="never" />
</menu>
```

MainActivity.java

```
package com.example.menubar;
import ...
public class MainActivity extends AppCompatActivity {
    View v;
    @Override
    protected void onCreate(Bundle i){
        super.onCreate(i);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem i){
        view = this.getWindow().getDecorView();
        switch(i.getItemId()){
            case R.id.red : view.setBackgroundResource(R.color.red);
                            return true;
            case R.id.gray: view.setBackgroundResource(R.color.gray);
                            return true;
            case R.id.green: view.setBackgroundResource(R.color.green);
                            return true;
            case R.id.blue: view.setBackgroundResource(R.color.blue);
                            return true;
            default:
                        return super.onOptionsItemSelected(i);
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu m){
        MenuInflater mi = getMenuInflater();
        mi.inflate(R.menu.main_menu, m);
        return true;
    }
}
```

15) Create an application that will change the color of text based on options selected from Context menu.

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        android:gravity="center">
    <TextView
        android:id="@id/text"
        android:text="Color changes here"
    />
    <Button
        android:id="@id/button
        android:text="Context menu"
    />
</LinearLayout>
```

floating_menu.xml → Should Create this menu

```
<?xml version="1.0" encoding="utf-8"
<menu xmlns:...>
    <item
        android:id="@+id/red
        android:title="red"/>
    <item
        android:id="@+id/blue"
        android:title="blue"/>
    <item
        android:id="@id/green"
        android:title="green"/>
    <item
        android:id="@id/yellow"
        android:title="yellow"/>
</menu>
```

MainActivity.java

```
package com.example.contextmenu;
import ...
public class MainActivity extends AppCompatActivity {
    private TextView t;
    @Override
    protected void onCreate(Bundle si){
        super.onCreate(si);
        setContentView(R.id.activity_main);
        Button b = findViewById(R.id.button);
        t = findViewById(R.id.text);
        registerForContextMenu(b);
    }
    @Override
    public void onCreateContextMenu(ContextMenu m, View v,
                                    ContextMenu.ContextMenuInfo
                                    info){
        super.onCreateContextMenu(m, v, info);
        MenuInflater i = getMenuInflater();
        i.inflate(R.menu.floating_menu, m);
    }
    @Override
    public boolean onContextItemSelected(MenuItem item){
        switch(item.getItemId()){
            case R.id.red : t.setTextColor(Color.RED);
                            return true;
            case R.id.yellow : t.setTextColor(Color.YELLOW);
                               return true;
            case R.id.green : t.setTextColor(Color.GREEN);
                              return true;
            case R.id.blue : t.setTextColor(Color.BLUE);
                             return true;
```

```
        default:
            return super.onOptionsItemSelected(item);
        }
    }
}
```