**ANDROID**

1. **.XML-**Designing(UI)
2. **.java-**Coding(Buisness logic)

**Activity Life Cycle:**  In android each activity can be pass through 6 life stages.These stages of activity can be managed by using some pre defined function.There are mainly 6 pre-defined function is present in the lifecycle of activity.These function is atomatic called based on the current life stages of activity.

onCreate():When application is created.

onStart():when activity is visible to othe.

onPause():When activity is partially visible to the user.

onResume():When user start intraction with the activity.

onStop():When application runs on background.

on Destroy():when aactivity is ended.

-------------------------------------------------------------------------------------

**XML:** Pre-defined elements

Naming Convention:Pascal Naming

Button: Button

Text(String Message):TextView

Form Control(TextBox):EditText

--------------------------------------------------------------------------------------

**Element Properrties:** camel case

textSize

textColor

textFontStyle

--------------------------------------------------------------------------------------

**Layout of the activity:**

LinearLayout

RelativeLayout

ConstraintLayout

FrameLayout

TableLayout

//linear layout is used to print layout in a simple way the main function

//Match parent and

**Message print on app:-**

**Logcat:-//**Only Developer Can See this Message

**Toast:** Application

Toast-class

makeText(this,””,duration)-function

show() –that enables the Toast to be displayed

**Syntax:**

Toast.makeText

**TextView Print**

**-------------------------------------------------------------------------------------------**

**Bind click event to the button:**

1. **Call a function on click of button**

Create aa method on .java page of activity :

public void showMsg(View vi)

{

}

1. **Bind on Click**

**//1.**  Apply a id to the element.

//2.

**Exceptiobn Handling:**Exception is the run time error which caused termination of the app.Exceptions are some unwanted situation of programs that may intrupt the execution of the program.

Exceptions are generated because wrong entries in the program.

//Toast

**\***Firstly try block is executed,if there is no exception in try block then catch block does not exxecuted.

**\*** Multiple catch may present with one try block,if you want to handle different type of Exceptions differently.Catch should be present just after the block of try

**Syntax:**

try

{

//statement

}

catch**(**Exception class obj)

{

}

Catch(Exception\_class\_obj)

{

}

* A finally block may present with each try block .This finally block executes in each cases either try throws an error or not

**Implicit Intent:** Intent is used to transfer from one activity to another activity.

Implicit Intent is used to transfer control from one activity to another within same application.

Emplicit intent is used to transfer control from one activity to another in different applications.

**Ex:** Open camera on click of button ,open flashlight on click of button,open gallery,open whatsapp.

**Type of Intent:**

1. Implicit Intent
2. Explicit Intent

To open next activity from current in same application ,Implicit intent is used.Intent class is used to create object of implicit intent.

Intent obj=new Intent(Context:File\_name.this,java\_file\_name.class;

Ex:Move to Login Activity from register activity:

Intent intent=new Intent(Register.this,Login.class);

-------------------------------------------------------------------

To finally start the activity “startActivity()” method is used.Which takes parameter of a intent class obj:

startActivity(intent\_class\_obj);

ex:startactivity(intent);

------------------------------------------------------------------------------------------------

Transfer from activity to another with loaded data:

To pass some value with Intent putExtra() function is used.

In putextra function a value can be passed in form of key and value.

**To hide action bar from current Activity use**

**“getSupportActionBar().hide();”**

**//stroke is used for border and corner is used for to give radius to page**

**onStart()**

**onClick()**

**onResume()**

**onDestroy()**

**onBackpressed()**

**OnCreateOptionMenu**

**Menu.xml :**Create a menu xml in Res/menu directory.

Once create the menu you items and you can inflate this items in any activity.

<menu>

<item title=”Setting

**Create a loader in java:**

ProgressDialog progree=new ProgressDialog();

progress.setTitle(“Set Title Here”);

progress.serMessage(“Set Discription Here”);

progress.show();//Shows the ProgressDialog

progress.hide();//hide the ProgressDialog

**------------------------------------------------------------**

**Adapter:-** Adapter works as a bridge between a sample view and a layout. It is used to create a view that can be add in any layouts.

To bind views in ViewPager adapter are inherited by **FragmentStateAdapter** class.

FragmentStateAdapter is a abstract class.Abstract class may have abstract any non-abstract

type methods.

When a class have any one abstract method then it must should be declared as a abstract

class.When a classs inherit a abstract class ,it may have to implement is abstract method of

parent classs .

Abstract method are those methods which have only declaration not definition . Abstract

methods are defined within child class.

**Projects worK**

**Create a New Project :**

MainActivity:Tablayout with 2 tabs:My Profile,My Life

Create a ViewPager2 and replace 2 fragment(MyProfileFragment,MyLifeFragment)

**----------------------------------------------------------------------------------------------**

**Create aa new project with three tabs:**

Calculator,History,Saved Transactions

And create a ViewPager2 to replace three fragment(CallFragment,HistoryFragment,TransactionFragment)

Foreach loop is used to access each element of a Collection.Foreach can only be applied on a collection like Array,List,ArrayList,HashMap etc.

For each loop access each element from start endex to the last.

**Syntax:-**

For(data\_type variable\_name:collection)

{

}

**Ex :** public static void main (string args[])

{

Int[] marks=new int[]{20,30,40,50};

For(int:marks)//

{

System.out.println(i);1

System.out.println(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

}

}

**Adapter For RecyclerView:**

To create a Adapter for recycler View,java class is inherited from the Abstract class called RecyclerView.Adapter:

onCreateView()

//inflate a design and return thedesign

{

}

Class ViewHolder extends RecyclerView.viewHolder{

Public ViewHolder(View itemView)

{

Return (super.itemView);

//Get Refrence of all recyclerViewItems

}

}

}