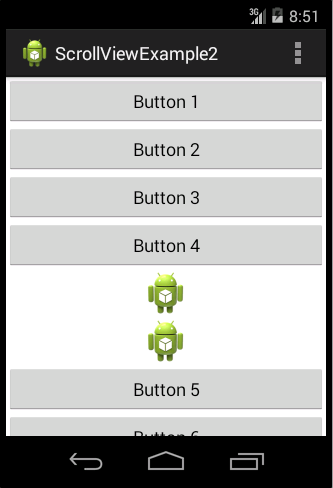
**Practical 5 (Fragment and Layout)**

**Question 1 (ScrollView)**

1. In the /layout directory, open the **activity\_main.xml** and design the interface as below using graphical layout. You should apply the **ScrollView** in the design. You may use any image as you wish. In this layout, you have 8 buttons and 2 images.



1. Your xml file should have something similar to xml below: **(p/s: You do not need to follow the xml as it might be different in each design.)**

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

<ScrollView xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"* >

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:orientation=*"vertical"* >

<Button

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 1"* />

<Button

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 2"* />

<Button

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 3"* />

<Button

android:id=*"@+id/button3"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 4"* />

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:src=*"@drawable/ic\_launcher"* />

<ImageView

android:id=*"@+id/imageView2"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:src=*"@drawable/ic\_launcher"* />

<Button

android:id=*"@+id/button4"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 5"* />

<Button

android:id=*"@+id/button2"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 6"* />

<Button

android:id=*"@+id/button1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 7"* />

<Button

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Button 8"* />

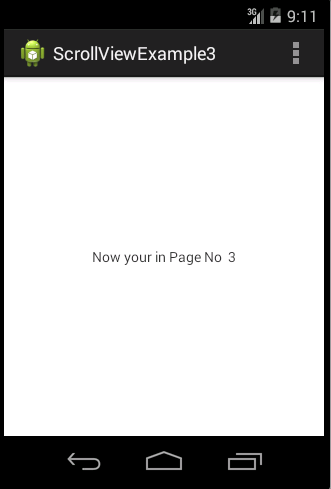
</LinearLayout>

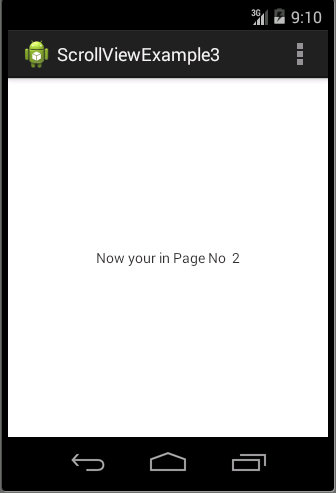
</ScrollView>

1. After make sure all the coding appropriate done, you may compile and run your coding .

**Question 2 (ViewPager)**

1. In this question, you are required to create an app with **ViewPager**. You should follow each step and understand the working of **ViewPager** by using **PagerAdapter**.





1. Create a file called pages.xml in **/res/layout** folder.

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

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:id=*"@+id/rel"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"* >

<TextView

android:id=*"@+id/pagenumber"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerHorizontal=*"true"*

android:layout\_centerVertical=*"true"* />

</RelativeLayout>

1. In the activity\_main.xml, provide the following coding.

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

tools:context=*".MainActivity"* >

<RelativeLayout

android:id=*"@+id/relativeTextview"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:layout\_below=*"@+id/header"*

android:padding=*"5dp"* >

<android.support.v4.view.ViewPager

android:id=*"@+id/reviewpager"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"* />

</RelativeLayout>

</RelativeLayout>

1. Create a java class under **/src** folder called ViewPagerAdapter. Use the following coding.

**public** **class** ViewPagerAdapter **extends** PagerAdapter

{

**int** size;

Activity act;

View layout;

TextView pagenumber;

Button click;

**public** ViewPagerAdapter(MainActivity mainActivity, **int** noofsize) {

// **TODO** Auto-generated constructor stub

size = noofsize;

act = mainActivity;

}

@Override

**public** **int** getCount() {

// **TODO** Auto-generated method stub

**return** size;

}

@Override

**public** Object instantiateItem(View container, **int** position) {

LayoutInflater inflater = (LayoutInflater) act

.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);

layout = inflater.inflate(R.layout.***pages***, **null**);

pagenumber = (TextView) layout.findViewById(R.id.***pagenumber***);

**int** pagenumberTxt=position + 1;

pagenumber.setText("Now your in Page No " +pagenumberTxt );

((ViewPager) container).addView(layout, 0);

**return** layout;

}

@Override

**public** **void** destroyItem(View arg0, **int** arg1, Object arg2) {

((ViewPager) arg0).removeView((View) arg2);

}

@Override

**public** **boolean** isViewFromObject(View arg0, Object arg1) {

// **TODO** Auto-generated method stub

**return** arg0 == ((View) arg1);

}

@Override

**public** Parcelable saveState() {

**return** **null**;

}

}

1. In the **MainActivity**, code the following source code.

**public** **class** MainActivity **extends** Activity {

**int** noofsize = 6;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

ViewPagerAdapter adapter = **new** ViewPagerAdapter(MainActivity.**this**,

noofsize);

ViewPager myPager = (ViewPager) findViewById(R.id.***reviewpager***);

myPager.setAdapter(adapter);

myPager.setCurrentItem(0);

}

@Override

**public** **boolean** onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.***main***, menu);

**return** **true**;

}

@Override

**public** **boolean** onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

**int** id = item.getItemId();

**if** (id == R.id.***action\_settings***) {

**return** **true**;

}

**return** **super**.onOptionsItemSelected(item);

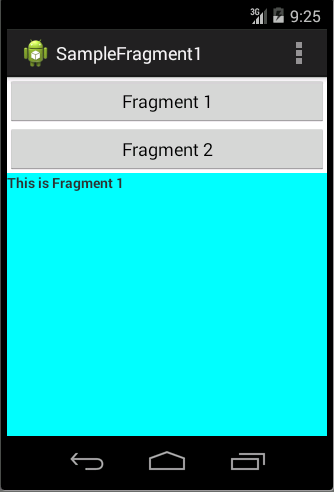
}

}

1. After make sure all the coding appropriate done, you may compile and run your coding .

**Question 3(Fragment)**

1. In the /layout directory, open the activity\_main.xml and design the interface as below using graphical layout or edit the xml file.



<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"* >

<Button

android:id=*"@+id/button1"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"selectFrag"*

android:text=*"Fragment 1"* />

<Button

android:id=*"@+id/button2"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"selectFrag"*

android:text=*"Fragment 2"* />

<fragment

android:id=*"@+id/fragment\_place"*

android:name=*"com.example.samplefragment1.FragmentOne"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"* />

</LinearLayout>

1. Create a file called fragment\_one.xml in /res/layout folder. Use the following settings for the UI.

<?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:background=*"@color/blue"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/textView1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:layout\_weight=*"1"*

android:text=*"This is Fragment 1"*

android:textStyle=*"bold"* />

</LinearLayout>

1. Create a file called fragment\_two.xml in /res/layout folder. Use the following settings for the UI.

<?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:background=*"@color/yelow"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/textView1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:text=*"This is Fragment 2"*

android:textStyle=*"bold"* />

</LinearLayout>

1. Create a class called FragmentOne.java under /src folder. Use the following source code.

**public** **class** FragmentOne **extends** Fragment

{

**public** View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle savedInstanceState)

{

//Inflate the layout for this fragment

**return** inflater.inflate(R.layout.***fragment\_one***, container, **false**);

}

}

1. Create a class called FragmentTwo.java under /src folder. Use the following source code

**public** **class** FragmentTwo **extends** Fragment

{

**public** View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle savedInstanceState)

{

//Inflate the layout for this fragment

**return** inflater.inflate(R.layout.***fragment\_one***, container, **false**);

}

}.

1. In the MainActivity.java, override the OnCreate() function as below:

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

}

**public** **void** selectFrag(View view)

{

Fragment fr;

**if**(view == findViewById(R.id.***button2***))

{

fr = **new** FragmentTwo();

}**else**

{

fr = **new** FragmentOne();

}

FragmentManager fm = getFragmentManager();

FragmentTransaction fragmentTransaction = fm.beginTransaction();

fragmentTransaction.replace(R.id.***fragment\_place***, fr);

fragmentTransaction.commit();

}

1. After make sure all the coding appropriate done, you may compile and run your coding .

**Question 4**

1. In this question, you are required to create an app to show your friends photo by clicking their name.



**Fragment2**

**Fragment1**

1. In this question, you should use 2 fragments. First fragment used to display the name list. Second fragment used to display friend’s photo.
2. You should provide at least 10 friend’s name in the listfragment.
3. When user clicks on the name, the photo of the name should be displayed on the Fragment 2.

**Question 5**

1. In this question, you are required to create an app to show the course name by clicking on the button.
2. In this question, you should create 2 fragment. Sample interface shows as below:
3. When user clicks on the button, the text on the right side should be changed.

