MOBILE APPLICATION DEVELOPMENT

LECTURE 6

Exercise 1:

Put a button and go to its properties, check layout width and layout height, change from wrap content to fill parent and/ or match parent, observe the change in layout of a button. Once done try the same with two buttons, change the text from Button to some special character like + or – and adjust the font size, type and color as well.

Exercise 1(B):

Drag 4 buttons to your main activity, adjust their length and width to be set equally on the screen, change the background color to Blue, Titles should be A, B, C and D.

Exercise 2:

From text field category, select password and add a button as well.

Goto java main activity and declare two private variables of type Button and EditText you may name them anything of your choice, for example:

//in main activity

private Button iunc;

private EditText north;

Now define a method that will display the contents in password field when user clicks the button:

You need to import toast library as well, that can be done as:

import android.widget.Toast;

public void myNewFunct( )

//add this method in onCreate method defined for MainActivity

{

//casting define variables

iunc = (Button) findViewById(R.id.button); //button here is the id of button

north = (EditText) findViewById(R.id.editText); //editText id the id of password field in layout

//set on click event on your button

iunc.setOnClickListener(new View.OnClickListener () {

public void onClick(View v){

//now you can display the nessage using toast here

Toast.makeText(

//name your mainactivity first

Main.Activity.this, north.getText(),

Toast.LENGTH.SHORT).show();

}

}

);

}

Exercise 3:

Let’s work with lists and check boxes.

From widgets, select couple of check boxes and a button by dragging them to your activity

Change the layout width to fill parent for each check box, rename the check boxes for list of items (anything may be fruits, vegetables, etc.)

Goto Java file, within MainActivity declare the variables for each of your check box (let’s assume we are working with 3) and button:

private Checkbox c1,c2,c3;

private Button b1;

//don’t forget to check the widgets checkbox and button, if not listed, do import the library as:

import android.widget.Checkbox;

import android.widget.Button;

Declare the method now:

For example:

//call this method in onCreate method as done for previous methods

/\*protected void onCreate (..){

..

myNewListener( );

}

\*/

public void myNewListener( )

{

//casting define variables

c1 = (CheckBox) findViewById(R.id.checkBox\_banana); //checkbox\_banana here is the id of firstcheckbox

c2 = (CheckBox) findViewById(R.id.checkBox\_apple); //checkbox\_apple here is the id of 2ndcheckbox

c3 = (CheckBox) findViewById(R.id.checkBox\_mango); //checkbox\_mango here is the id of 3rdcheckbox

b1 = (Button) findViewById(R.id.button); //button is the id of button

//now set new on click listener

b1.setOnClickListerner(new View.onClickListener(){

public void onClick(View v) {

//add buffer here

StringBuffer iunc= new StringBuffer( ) ;

//appending checkbox value

iunc.append(“banana : ” ).append(c1.isChecked( ));

iunc.append(“\napple : ” ).append(c2.isChecked( ));

iunc.append(“\nmango : ” ).append(c3.isChecked( ));

//adding toast to display value

Toast.makeText(MainActivity.this, iunc.toString( ), Toast.LENGTH\_LONG).show();

}

}

);

}

Run your application. Hope everything is working fine.

Exercise 4:

Now we will work on the method to display the value as soon as the user selects a checkbox, with out clicking at any button event, for this you need to define a method:

//call this method in onCreate method as done for previous methods

public void myCheckboxCheck() {

//cast the variables:

c1 = (CheckBox) findViewById(R.id.checkBox\_banana); //checkbox\_banana here is the id of firstcheckbox

c1.setOnClickListerner(new View.onClickListener(){

public void onClick(View v) {

if (((CheckBox)v).isChecked() ) {

Toast.makeText(MainActivity.this, “Banana is selected”, Toast.LENGTH\_LONG).show();

}

}

}

);

}

Exercise 5:

Develop and application that will be having 6 check boxes with titles of main category for any convertor (for example temperature, currency etc. ).Do check if it is returning true for selected categories or not.

Exercise 6:

Working with multiple activities:

Right click on resources folder (res) - > layout folder (right click) -> new -> activity -> blank activity

Name your activity -> click Finish

Go to Manifest folder o verify newly added activity xml, copy the tag <intent-filter> … </intent-filer> inside the second activity before closure. </activity>

Change the category from LAUNCHER to DEFAULT and name copy the path of package ( to find this, you may go to layout folder, select the newactivity.xml file, copy the path of your package from tag tools:context=”com…..” copy the path from com. Onwards go back to android manifest.xml and copy the path to <action android name = path”> instead of Main

Now try to link first activity with second activity.

Add a button to main activity rename it as you want it to be.

Go to Java folder main activity, cast the button : //inside MainActivity method

private static Button submit;

//call the following method in onCreate method as done for previous methods

public void goNextActivity() {

//casting variable of button

submit = (Button) findViewById(R.id.button); //button is the id of button

//adding listener to button

submit.setOnClickListerner(

new View.onClickListener(){

public void onClick(View v) {

//creating object of intent class

Intent intent = new Intent(“*path*”);

/\*as argument pass the path of action to find this, you may go to layout folder, select the newactivity.xml file, copy the path of your package from tag tools:context=”com…..” copy the path from com. onwards \*/

startActivity(intent);

}

}

);

}

Exercise 7:

Develop an application that will be having a button PROFILE, if selected by user; display your profile (including name, father’s name, email, contact, semester, GPA etc.)

Exercise 8:

From palettes select date and time and drag analog and digital clock to your activity, also a button to the same. Now go to java main activity file,

private static Button submit;

private static DigitalClock digi;

private static AnalogClock ana;

//call the following method in onCreate method as done in previous exercises

//define the method

public void myClockSwapApp() {

//cast your variables:

digi = (DigitalClock) findViewById(R.id.digitalClock); //digitalClock here is the id of Digital CLock

ana = (AnalogClock) findViewById(R.id.analogClock); //ana here is the id of Analog

submit = (Button) findViewById(R.id.button); //button is the id of button

//adding listener to button

submit.setOnClickListerner(

new View.onClickListener(){

public void onClick(View v) {

if(digi.getVisibility == DigitalClock.GONE) {

digi.setVisibility(DigitalClock.VISIBLE);

ana.setVisibility(AnalogClock.GONE);

} else {

digi.setVisibility(DigitalClock.GONE);

ana.setVisibility(AnalogClock. VISIBLE);

}

}

}

);

}