Lab 1 – First trial of the android SDK

Set up the android development environment

Please go to the following website and follow the description on the website to install the SDK.
http://developer.android.com/sdk/index.html
(Note: This link is for your reference, the sdk had already installed in the lab computers)

Task 1	Hello world version 1
Knowledge learn in this task:	
Function: "onCreate"	1) The first function called in the program 2) All coding in "onCreate" should be placed after the **super.onCreate(savedInstanceState);**
Procedure of the task:	
Step 1 Create project in android studio	

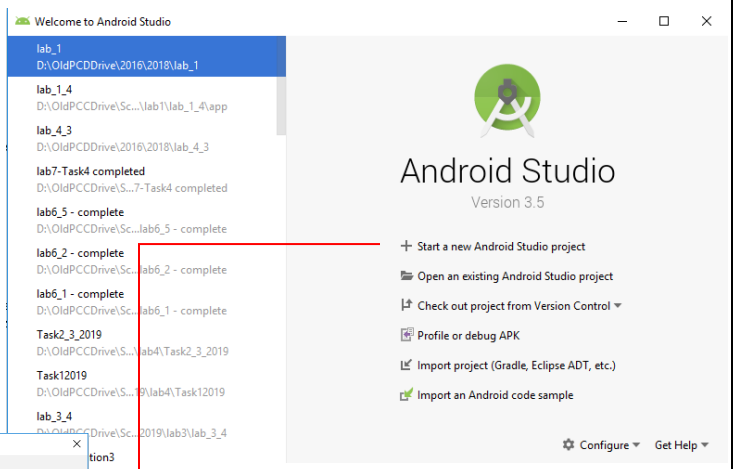
01) Click "All program->Android Studio->Android Studio"
➔ "Start a new Android Studio Project"

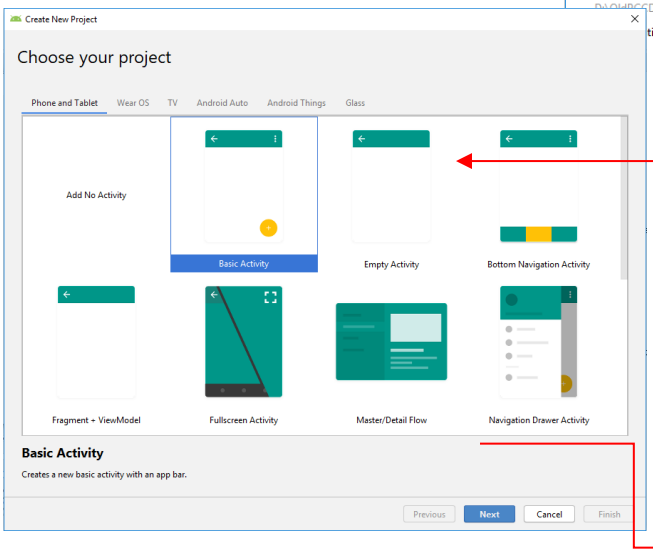
02) Enter the following information

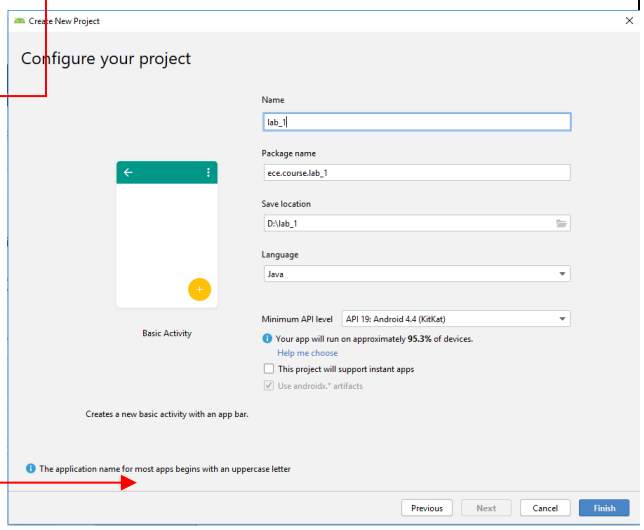
1) Select "Basic Activity" and Press "Next >"

2) Enter Application name
(You can use your own or using the "lab_1")
Now, package name is "**ece.course.lab_1**"

3) Select Minimum SDK to **"API 19: Android 4.4 (KitKat)" and "Java" as** programming language and Press "Finish>"

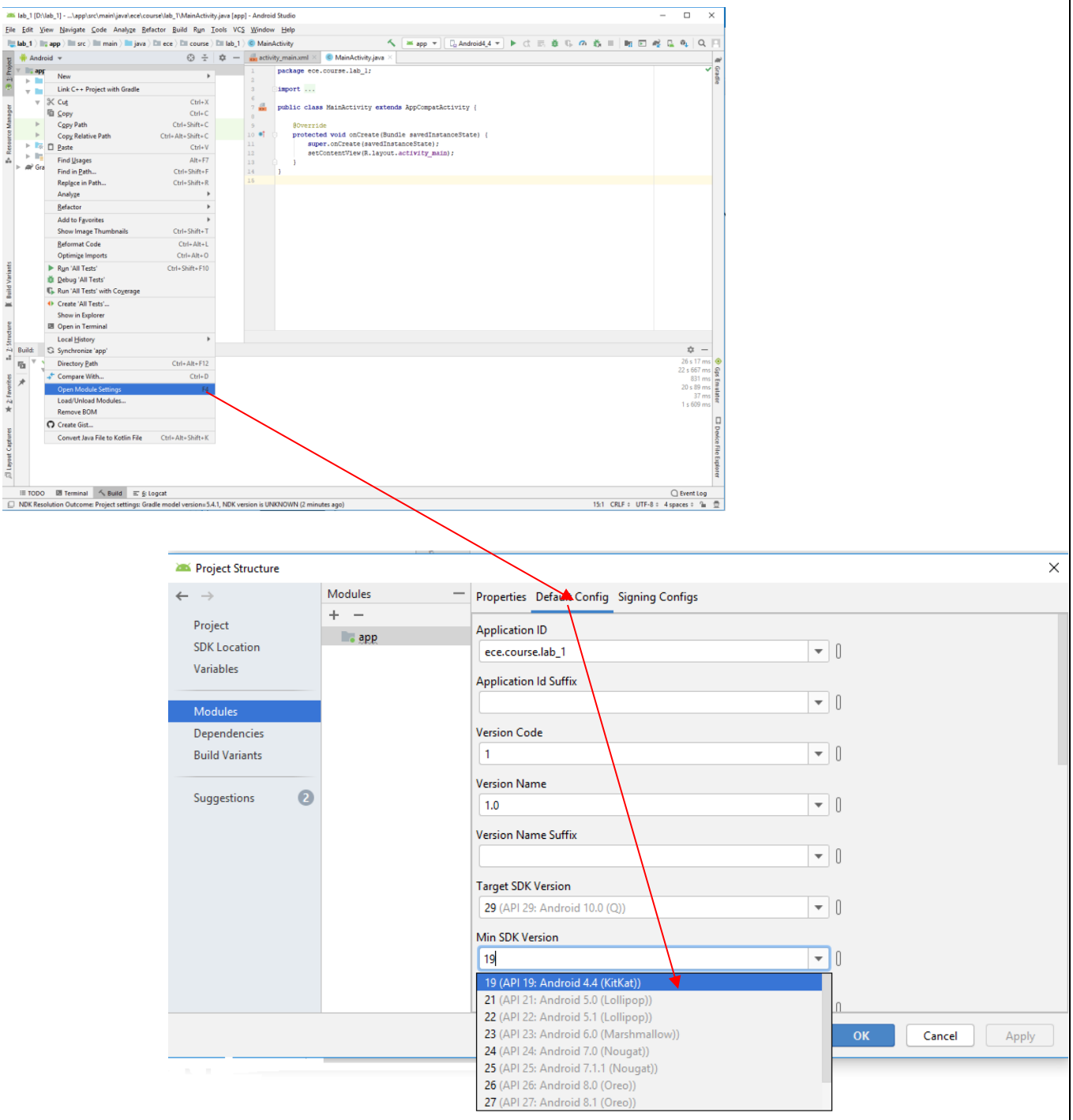






4) Configure build target version:

In app, Right Click->Open Module Settings; Choose Default Config -> Select Android 4.4

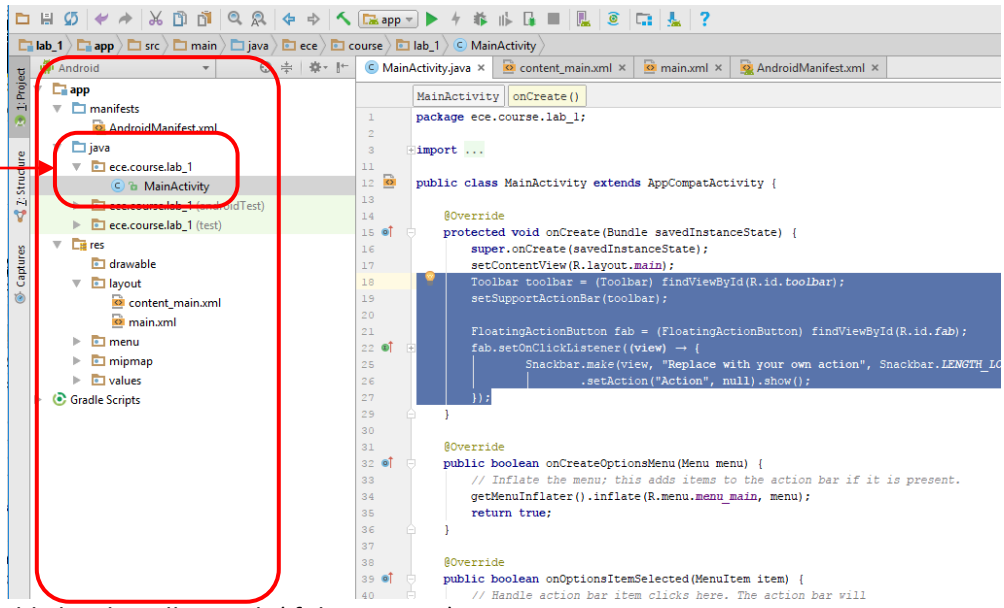


Choose “OK” and close it

Step 2

Set display by coding

01) Open the java file of the created activity



*you can find the file in "Package Explorer"

**The file is in <Project name>

➔ "src"

➔ <package name>

➔ <activity name>

***You can remove all the highlighted Toolbar code(if there is any)

02) Set up new "TextView" for displaying the message.

a) Create a new TextView

Code:

```
TextView textView = new TextView(this);
```

Class — Variable name — Constructing variable

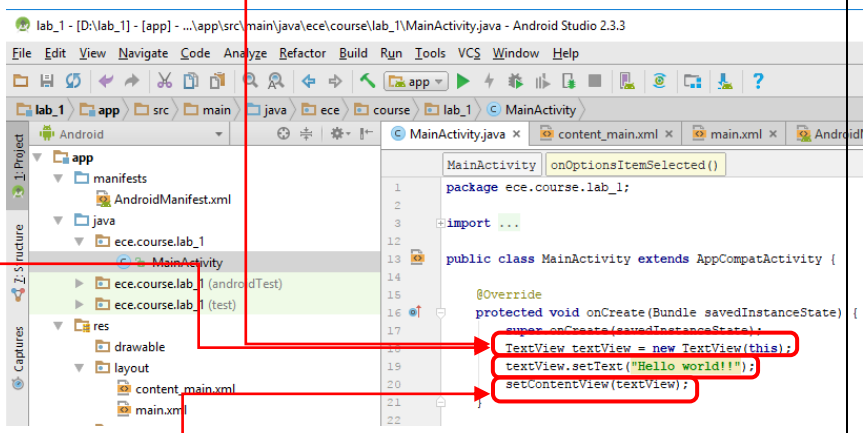
b) Set the TextView's text

Code:

```
textView.setText("Hello world!!");
```

TextView name — Text need to display

c) Change the display to showing the text



Code change:

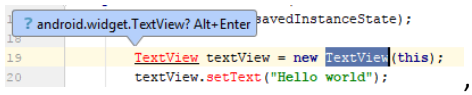
1) setContentView(R.layout.main);

➔ setContentView(textView);

TextView name

2) In res->layout->main.xml, remove (if there is any)

```
<include layout="@layout/content_main" />
```

*If you see error picture like this 

, click on it and select “Alt+Enter” to import the library

Notice:

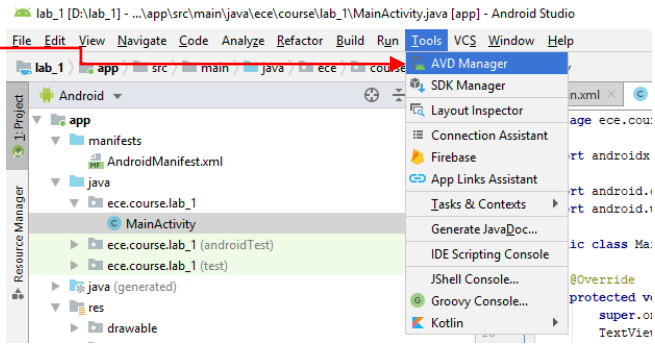
3) You may see some other codes like “onCreateOptionMenu”,”onOptionsItemSelected” in the java file, these codes are not being used in this lab and you can ignore it
4) The toolbar code also not being used, you can ignore it or just delete the toolbar code

Step 3

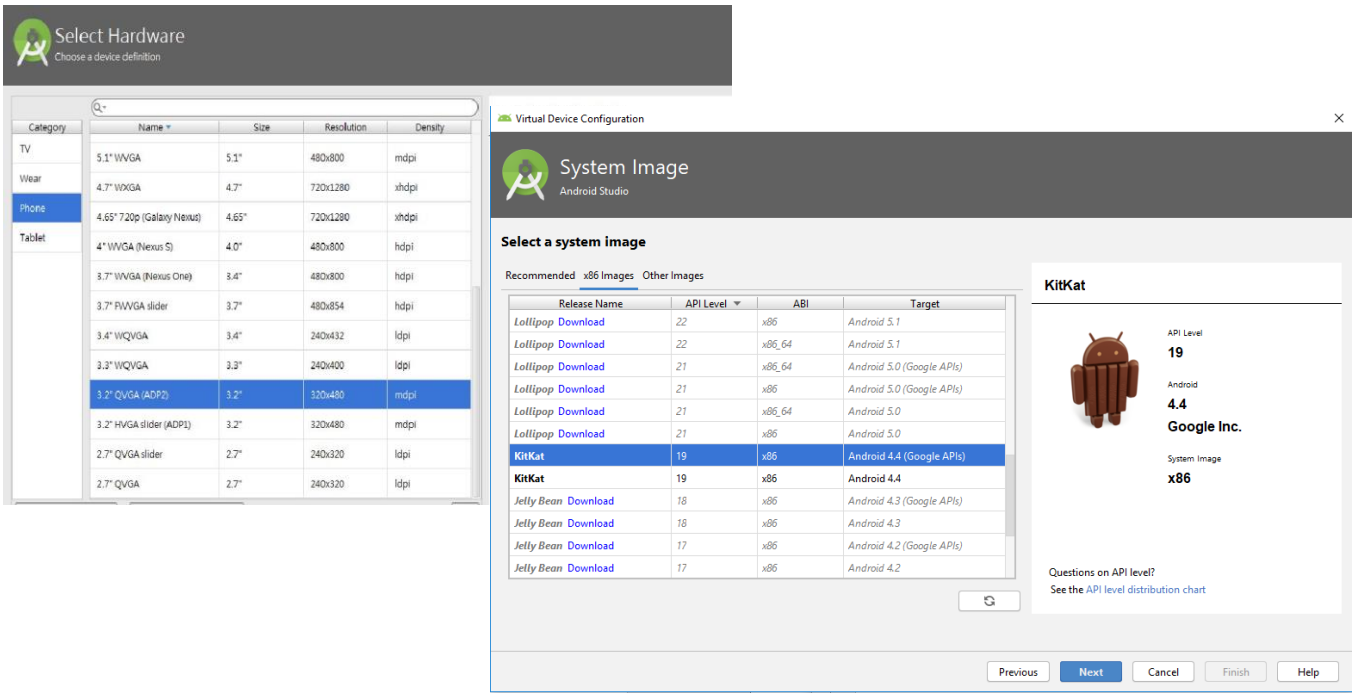
Test the apps by emulator

For the first time to use emulator, you need to do some configuration.

a) Open Android Virtual Device Manager

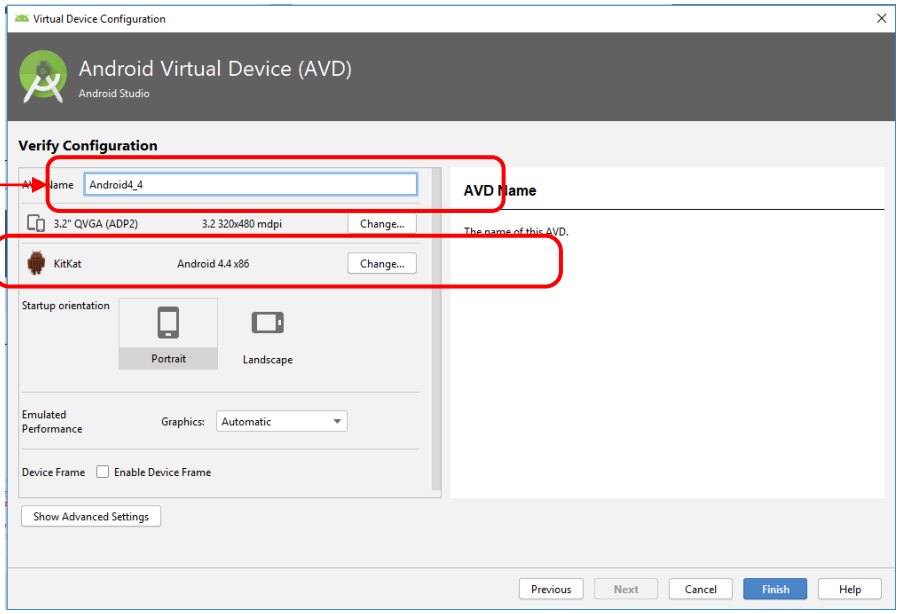


b) Click “+Create Virtual Device”



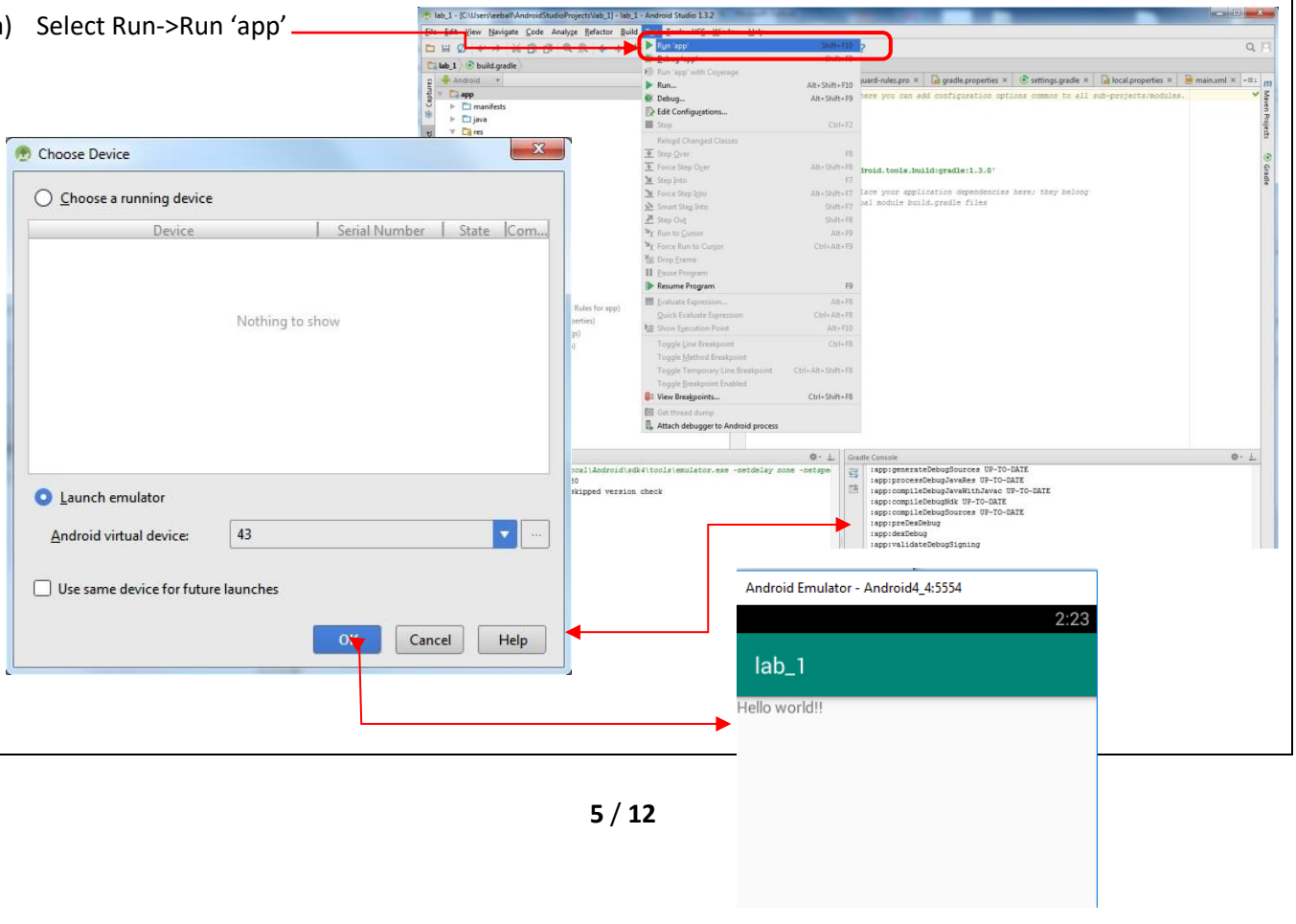
c) Enter the name
(You can use “Android_4.4” or think of you own name)

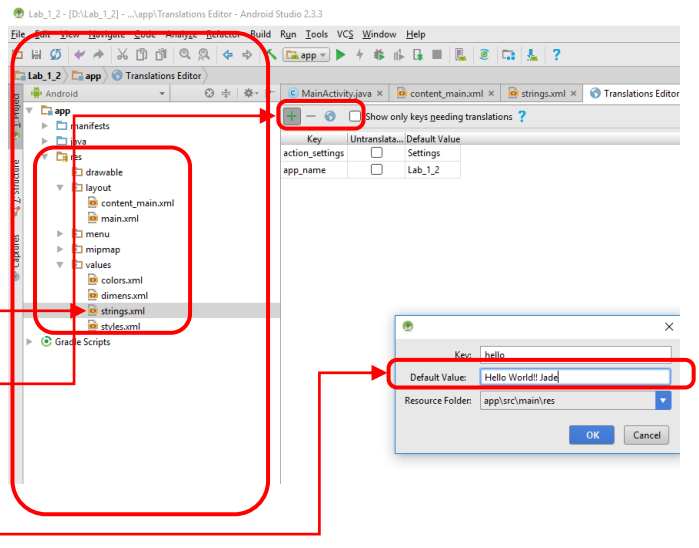
d) Select the Target
(Choose “Android 4.4”, since the lab is use Android 4.4 as the build target)



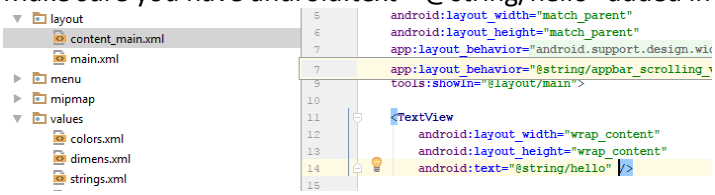
Select run/debug on the emulator

a) Select Run->Run ‘app’



Task 2	Hello world version 2
Knowledge learn in this task:	
“res” ➔ “values” ➔ “strings.xml”	1) It is a file for storing some useful values. 2) It is an editable file.
Procedure of the task:	
Step 1 Create project in android studio	
Use the method in Task 1 to create a new Project Project name: Lab_1_2 (or your own one) Target SDK: Android 4.4 Package name: ece.course.lab_1_2	
Step 2 Set display by xml	
01) Open the “strings.xml” *The file is in <Project name> ➔ “res” ➔ “values” ➔ “strings.xml” 02) Add an element “hello” 03) Change value to “Hello world!!” + your name	

04) Make sure you have android:text="@string/hello" added in content_main.xml



(Notes: Make sure you are using setContentView(R.layout.main) instead of setting focus to textview)

Step 3

Test the apps by emulator

Task 3	Set up UI and try to use different views

Knowledge learn in this task:

"res" ➔ "layout"	1) A place to store the user's layout 2) All layout is stored in xml format
"res" ➔ "drawable"	1) A place to store the image
ImageView	1) A kind of View, which use to display image 2) You can custom what image it will display.

Procedure of the task:

Step 1

Open the project of last task

Step 2

Set the UI

01) Set the resources used in the layout

a) Open “strings.xml”

b) Add new string

1) Click “Open editor”
2) Click “Add…”
3) Enter “Key”

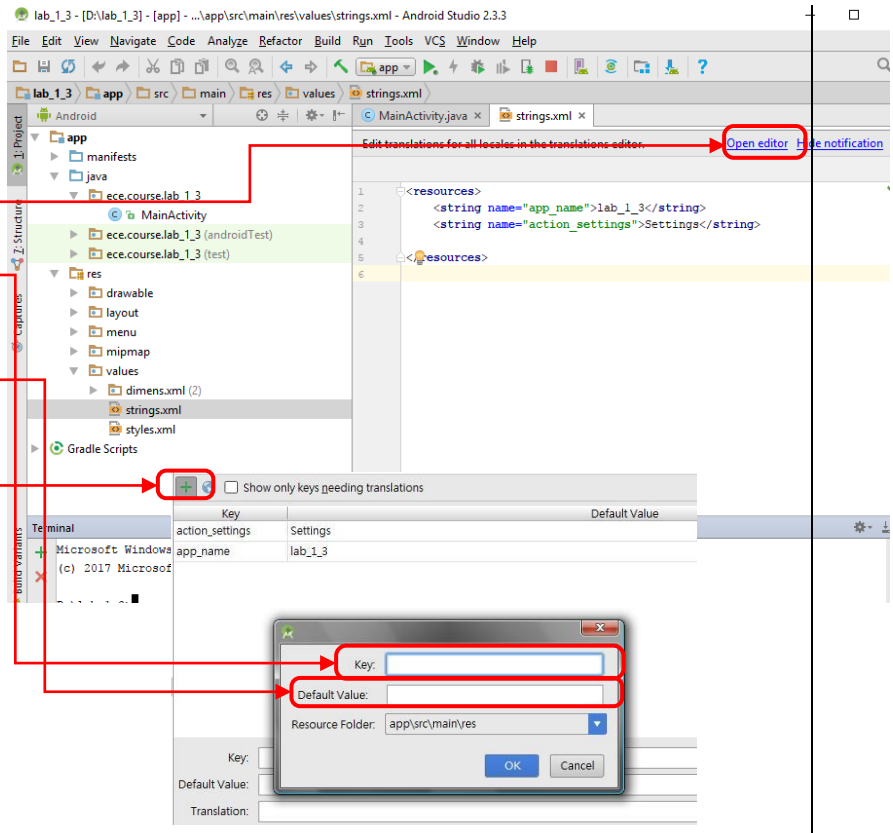
(This time enter “nameBoy” as the “Key”)

4) Enter “Default Value”

(This time enter “A boy” as the “Default Value”)

c) Add one more String
Name: nameGirl
Value: A girl

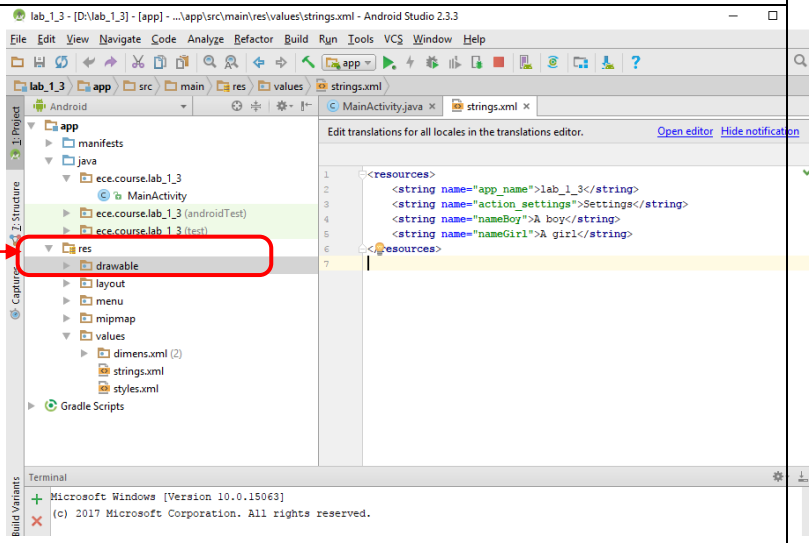
d) Save the file



02) Open the drawable folder

a) Dig the picture “boy.jpg” into the folder

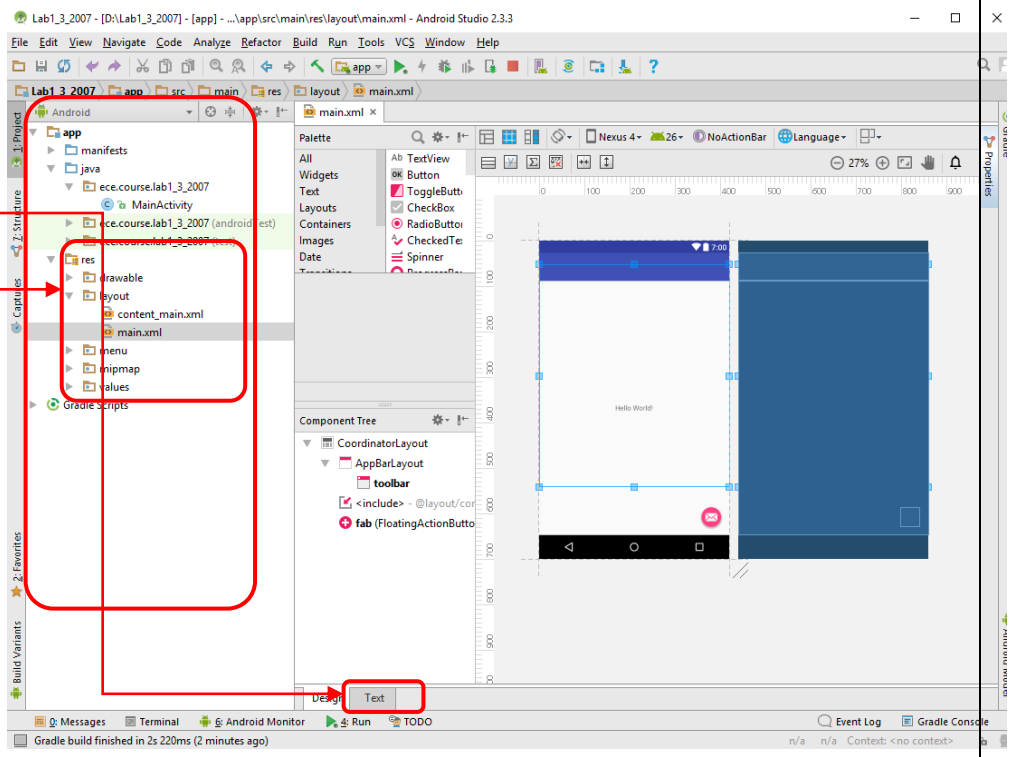
b) Dig the picture “girl.jpg” into the folder



03) Set the layout

a) Open the “main.xml” under “layout”

b) Open the text editor mode by click the tab under the editor



c) Delete all the tags in main.xml, For example, if you saw “Hello World” tag, **delete it**

```
<TextView
        android:layout_width=”wrap_content”
        android:layout_height=”wrap_ content”
        android:text=”Hello World!”
        />
```

d) Add a new TextView tag in the LinearLayout tag

```
<TextView android:id=”@+id/tvName”
        android:layout_width=”fill_parent”
        android:layout_height=”wrap_content”
        android:textSize=”20sp”
        android:text=”@string/nameBoy” />
```

e) Add a ImageView tag under the TextView tag in the LinearLayout tag

```
<ImageView android:id=”@+id/ivPicture”
        android:layout_width=”fill_parent”
        android:layout_height=”wrap_content”
        android:layout_weight=”1”
        android:src=”@drawable/boy” />
```

f) Add a LinearLayout tag under the ImageView tag in the LinearLayout tag

```
<LinearLayout android:layout_width=”fill_parent”
        android:layout_height=”wrap_content” >
</LinearLayout>
```

g) Add a EditText tag inside the LinearLayout tag (the inner one created by you)

```
<EditText android:id=”@+id/etName”
        android:maxLines=”1”
        android:layout_width=”wrap_content”
        android:layout_height=”wrap_content”
        android:layout_weight=”1” />
```

h) Add a Button tag inside the LinearLayout tag (the inner one created by you)

```
<Button android:id=”@+id/btnName”
        android:text=”Change”
        android:layout_width=”wrap_content”
        android:layout_height=”wrap_content” />
```

i) Add a Button tag under the LinearLayout tag (the inner one created by you)

```
<Button android:id=”@+id/btnPicture”
        android:text=”Change Picture”
        android:layout_width=”fill_parent”
        android:layout_height=”wrap_content” />
```

Make sure item d) to i) above should be put in the cursor position as below:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">


</LinearLayout>
```

Notice that we are using LinerLayout, if you are using Relativelayout, you have to make use of android:layout_below properties

Step 3

Test the apps by emulator

Task 4	An app that can change picture by one button and can change the name of the picture.
Knowledge learn in this task:	
Button	1) A kind of View, that design act as a button 2) It uses with the OnClickListener
EditText	1) A kind of View, that design for user to input text 2) You can use it to obtain text input from the user.
OnClickListener	1) It mainly uses with button. 2) What will perform in the button click is the coding under the function onClick in it.
Procedure of the task:	
Step 1 Open the project of last task	
Step 2 Set the link between the coding and the display	

01) Open java file of the activity

02) Start coding in the function “onCreate”

03) Implements the “Change Picture” button

a) Link the Button “btnPicture”

Code:

```
Button btnPicture = (Button) findViewById(R.id.btnPicture);
```

Class — Variable name — Class wanted to change to — Function to find target View — The Target ID

Should be same

Must be same

```
<Button android:id="@+id/btnPicture"
        android:text="Change Picture"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
```

b) Implements the onClick events, the logic need to be implemented by you.

Code:

```
btnPicture.setOnClickListener(new OnClickListener() {
        public void onClick(View view) {
        }
});
```

Variable name — Code placement for the onClick events — “onClick” function — Set OnClickListener

c) Implements a variable for determinant what should display

Code:

```
boolean isBoy = true;
```

Class: boolean | Variable name: isBoy | Initialize value: true

```
public class ******* extends Activity {
    boolean isBoy = true;
    @Override
    Public void onCreate(Bundle savedInstanceState) {
        Super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

d) Link the TextView "tvName" (Put in the onClick Events)

Code:

```
TextView tvName = (TextView) findViewById(R.id.tvName);
```

e) Change the text of "tvName" (Put in the onClick Events)

Code:

For display boy case:

```
tvName.setText(R.string.nameBoy);
```

For display girl case:

```
tvName.setText(R.string.nameGirl);
```

Must be same

String for boy
String name: nameBoy

String for girl
String name: nameGirl

f) Link the ImageView "ivPicture" (Put in the onClick Events)

Must be same

Code:

```
ImageView ivPicture = (ImageView) findViewById(R.id.ivPicture);
```

g) Change the picture of "ivPicture" (Put in the onClick Events)

Code:

For display boy case:

```
ivPicture.setImageResource(R.drawable.boy);
```

For display girl case:

```
ivPicture.setImageResource(R.drawable.girl);
```

Must be same

Picture for boy
Picture name: boy.jpg

Picture for girl
Picture name: girl.jpg

Must be same

*It is better, if the coding is placed after **setContentView(R.layout.main);**

04) Implements the "Change Name" button

a) Link the Button "btnName"

b) Implements the onClick events

c) Link the EditText "etName"

Code:

```
EditText etName = (EditText) findViewById(R.id.etName);
```

d) Get the Text in "etName"

Code:

```
String name = etName.getText().toString();
```

e) Link the TextView

f) Change the "tvName" text

Step 3

Test the apps by emulator