## block 1

**TOTAL POINTS 14**

1.Question 1

Which folder of your Android Studio project holds the XML files that describe the graphical interfaces of the MyApp application?

**1 / 1 point**



MyApp/app/gui



MyApp/app/values



MyApp/app/manifests



MyApp/app/res/layout



MyApp/app/res/drawable



MyApp/app/java

**Correct**

2.Question 2

You have defined a graphical user interface in an XML layout file named myUI. You create a new activity in a file name MyActivity.java. How to you specify that MyActivity displays myUI?

**1 / 1 point**



In the onCreate method of MyActivity.java I use ""R.id.myUI.display();"



It's impossible. Since the activity is named MyActivity, the system will try to load a layout named "activity\_my.xml".



In the onCreate method of MyActivity.java I call the setContentView method and I provide "R.layout.myUI" as parameter.



In the onCreate method of MyActivity.java I call the setContentView method (without parameter) and in the root element of the myUI.xml file I indicate android:activity="MyActivity".



In the onCreate method of MyActivity.java I call the onStart method and I provide "R.id.myUI" as parameter.



In the onCreate method of MyActivity.java I use "myUI.display();"

**Correct**

3.Question 3

The visual structure of the user interface can be defined...

**1 / 1 point**



exclusively in the Java code, by creating objects extending View or ViewGroup.



exclusively in an XML file listing the different Views and ViewGroups



either in the Java code or in an XML file but not in both at the same time for a single "screen".



in the Java code and/or in an XML file

**Correct**

4.Question 4

What could be the interest of creating a button (or any other widget) from the Java code of an activity rather than from the associated XML layout file?

**1 / 1 point**



It requires fewer lines of code.



It makes it possible to associate with the button a text which changes according to the language settings of the device.



It makes it possible to create the button only if a given condition is fulfilled at runtime.

**Correct**

5.Question 5

What could be the interest of creating a button (or any other widget) from an XML layout file rather than from the Java code of an activity?

**0.667 / 1 point**



The design view allows for a preview of the result without the need to deploy the app on a (real or emulated) device.

**Correct**



It separates the visual design and presentation aspects of the application from its control. Each aspect can even be taking care of by specialized persons.



It makes the display of the application faster.

You didn’t select all the correct answers

6.Question 6

Name the class which represents pop-up messages.

**1 / 1 point**

Toast

**Correct**

7.Question 7

In addition to the activity or application context, and the message to display, which parameter does the makeText method from the Toast class take?

**1 / 1 point**



the duration of the pop-up (Toast.LENGTH\_SHORT or Toast.LENGTH\_LONG)



the position of the pop-up (Toast.TOP, Toast.CENTER, or Toast.BOTTOM)



the language of the message (Locale.ENGLISH, or Locale.FRENCH...)



the background color of the pop-up (Color.BLUE, Color.BLACK...)



the priority of the message (Toast.URGENT or Toast.NON\_URGENT)

**Correct**

8.Question 8

Which method from the Toast class allows you to display a pop-up message you've prepared?

**1 / 1 point**



show()



sendToFront()



display()



bringToForeground()

**Correct**

9.Question 9

In an XML layout file you specify that "onClick" a button must call the "doSomething" method. In the activity Java file you create a method "public void doSomething(View v)". What does the 'v' parameter refer to?

**1 / 1 point**



It indicates the view which triggered the method call.



It refers to the graphical preferences set by the user.



It points to the layout onto which the button is positioned.

**Correct**

10.Question 10

Which method of the Activity class allows you to retrieve a reference to a widget defined in the XML layout file (assuming you know the "name" of the widget)?

**0 / 1 point**

findViewById(R.id.name)

**Incorrect**

11.Question 11

Consider the following code:



1

2

3

4

5

6

7

8

9

10

11

12

13

14

package fr.centralesupelec.galtier.tempo;

import android.graphics.Color;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button myButton = (Button)findViewById(R.id.myBbutton);

myButton.-------------(Color.BLUE);

}

}

Which method must be called at the location indicated with "-----" to turn the background of the button to blue.?

**1 / 1 point**

setBackgroundColor

**Correct**

12.Question 12

The following code displays a pop-up message reading "hello" for a short duration:



1

Toast.makeText(getApplicationContext(), "hello", Toast.LENGTH\_SHORT);

**1 / 1 point**



true



false

**Correct**

13.Question 13

Which of the following code extracts create a button which displays a pop-up message when clicked?

**1 / 1 point**





1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

package fr.centralesupelec.galtier.tempo;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void performAction (View v) {

Toast.makeText(getApplicationContext(),

"clicked!",

Toast.LENGTH\_SHORT).show();

}

}



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

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

<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="fr.centralesupelec.galtier.tempo.MainActivity">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="click me!"

android:id="@+id/myBbutton"

android:layout\_alignParentTop="true"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:onClick="performAction" />

</RelativeLayout>

**Correct**





1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

package fr.centralesupelec.galtier.tempo;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button myButton = (Button)

findViewById(R.id.myBbutton);

myButton.setOnClickListener(

new View.OnClickListener() {

public void onClick(View v) {

Toast.makeText(getApplicationContext(),

"clicked!",

Toast.LENGTH\_SHORT).show();

}

});

}

}



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

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

<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="fr.centralesupelec.galtier.tempo.MainActivity">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="click me!"

android:id="@+id/myBbutton"

android:layout\_alignParentTop="true"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true" />

</RelativeLayout>

**Correct**





1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.RelativeLayout;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button myButton =

new Button(getApplicationContext());

myButton.setText("click me!");

myButton.setOnClickListener(

new View.OnClickListener() {

public void onClick(View v) {

Toast.makeText(getApplicationContext(),

"clicked!",

Toast.LENGTH\_SHORT).show();

}

});

RelativeLayout layout = (RelativeLayout)

findViewById(R.id.myLayout);

layout.addView(myButton);

}

}



1

2

3

4

5

6

7

8

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

<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"

android:id="@+id/myLayout"

tools:context="fr.centralesupelec.galtier.tempo.MainActivity">

</RelativeLayout>

**Correct**

14.Question 14

For how long (in second) does a Toast displayed for a duration "Toast.LENGTH\_SHORT" stays on the screen?

**1 / 1 point**

2

**Correct**

and LENGTH\_LONG is 3.5 seconds.See here:

<http://stackoverflow.com/questions/7965135/what-is-the-duration-of-a-toast-length-long-and-length-short>