





Unit 1 – ICS4U0 – Android Interfaces

Sample Test - December 20, 2022

Name: Solutions

Total	%	Knowledge 	Communication 	Thinking 	Application 
(78)	%	(16)	(21)	(19)	(22)

Knowledge

1. Identify the type of View shown in each picture.

Interesting



(a) TextView

COUNT UP!



(b) ImageView

(c) Button

Email

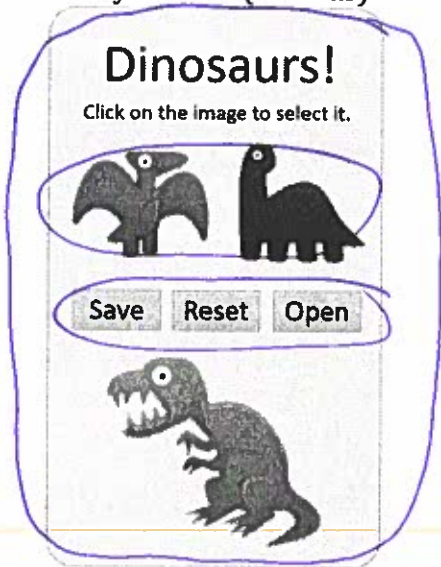
(e) EditText

(d) ImageView



(f) ImageView

2. Outline the LinearLayouts needed around the views they enclose. (3 marks)



3. Fill in the hex codes for each colour. (5 marks)

Red	<u>#FF0000</u>
Yellow	<u>#FFFF00</u>
Green	<u>#00FF00</u>
Cyan	<u>#00FFFF</u>
Blue	<u>#0000FF</u>
Magenta	<u>#FF00FF</u>
Black	<u>#000000</u>
White	<u>#FFFFFF</u>

4. Circle the valid Android image names. *No capitals for image names*

Dinosaur

dinosaur

DINOSAUR

ldino

oneDino

dinol

5. Fill in this code:

You have 5 points android:id="@+id/score" change to: You have lost the game.	TextView <u>score</u> = (TextView) findViewById(R.id. <u>score</u>); <u>score</u> .setText("You have lost the game");
Name android:id="@+id/first" get the text and clear the EditText.	EditText <u>first</u> = (EditText) findViewById(R.id. <u>first</u>); String name = <u>first</u> .getText().toString(); <u>first</u> .setText("");

Communication

6. Select true or false for each statement.

/6

- T a) The property used to link a button with a method in java is an id. *onclick*
- T b) dp stands for dimension-independent pixel. *density*
- T c) Android screens are laid out using XML and their actions are coded using java.
- T d) An onClick attribute can be added to an ImageView.
- T e) Android view ids can have capitals.
- T f) An example of a ViewGroup is a Parent. *LinearLayout*

7. Fill in the appropriate term that matches the description in the front column.

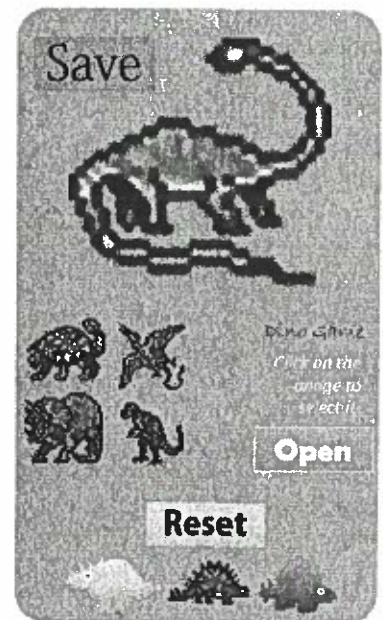
/5

<i>Pixelation</i>	(a) An image error where blocky pixels appear.
<i>View view</i>	(b) The parameter of the method associated with an onClick.
<i>Linear layout</i>	(c) A ViewGroup with an orientation attribute.
<i>View</i>	(d) An item that is added to an app screen for the user to manipulate.
<i>onClick</i>	(e) A property that is needed if the View can be clicked.

8. Name each PARC principle and explain how each is broken by this app.

/8

<i>Proximity</i>	<i>Similar items (like save, open, reset buttons) should be grouped together and separated with a space.</i>
<i>Alignment</i>	<i>Save + 4 pics are left aligned, large dino + reset is centered, the instructions + open are right aligned. There should be <u>one</u> alignment, not 3.</i>
<i>Repetition</i>	<i>Many different font; there should be a max of 2. 2 types of pictures - pixelated and bottom ones; there should be 1.</i>
<i>Contrast</i>	<i>The Reset button stands out, but why? Contrast should be used for title (dino name possibly) to draw eye there first.</i>



9. Define inflation and describe why it is useful in Android development.

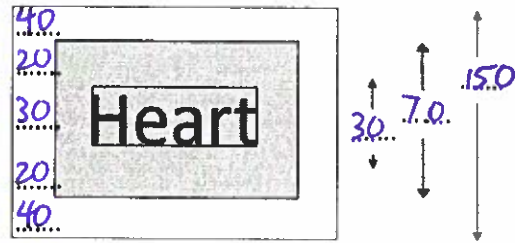
/2

- ① → Inflation is the process of translating XML to java. It is coded in the onCreate method in the java file.
- ② → Inflation is useful because it allows us to set up screens in XML (which is easier) and code in java (which has full functionality). Thus, we can get the best of both worlds.

Thinking

10. Using the code, fill in the dimensions of the button. /3

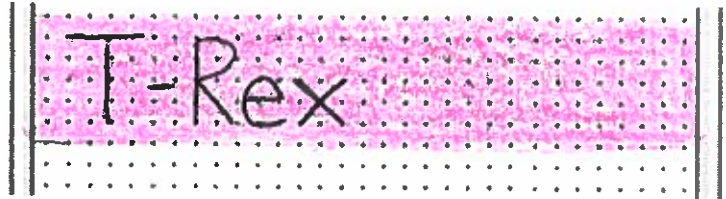
```
<TextView
    android:text="Heart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30sp"
    android:padding="20dp"
    android:layout_margin="40dp"/>
```



11. Draw these widgets using colour. The width of the screen is shown. Each dot square is 10 dp. /9

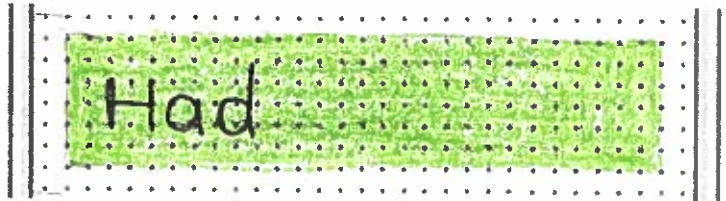
(a)

```
<TextView
    android:text="T-Rex"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="40sp"
    android:background="#FF00FF"
    android:padding="10dp"/>
```



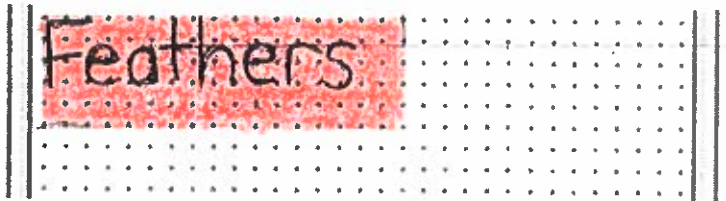
(b)

```
<TextView
    android:text="Had"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:background="#00FF00"
    android:padding="20dp"
    android:layout_margin="10dp"/>
```



(c)

```
<TextView
    android:text="Feathers"
    android:layout_width="170dp"
    android:layout_height="50dp"
    android:textSize="30sp"
    android:background="#FF0000"/>
```



12. Circle and correct seven errors in this code. /7

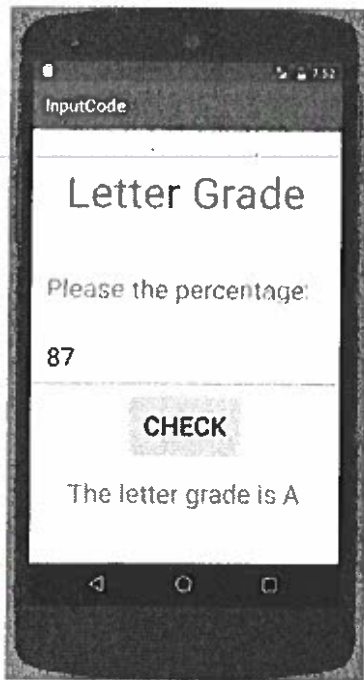
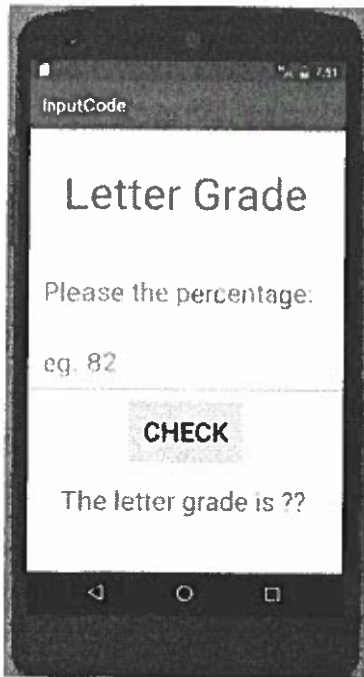
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_presents" parent
    android:layout_height="match_parent"
    android:orientation="striped_seawater"> horizontal
    <EditText
    <textField
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Date"
        android:textSize="20pixels" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10" dp
        android:text="Done"
        android:textSize="20sp" />
```

<LinearLayout>

Application

13. Fill in the blanks to create the below app. Also fill in the java for the onClick:

/12



Percent	Grade
90+	A+
80-89	A
70-79	B
60-69	C
50-59	D
49-	F

```
<?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:orientation="vertical" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="40dp"
        android:text="Letter Grade"
        android:textSize="50sp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="20dp"
        android:text="Please the percentage:"
        android:textSize="30sp" />
    <EditText
        android:id="@+id/input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="eg. 82"
        android:padding="20dp"
        android:textSize="30sp" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:onClick="clicked"
        android:padding="20dp"
        android:text="check"
        android:textSize="30sp" />
    <TextView
        android:id="@+id/output"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="20dp"
        android:text="The letter grade is ??"
        android:textSize="30sp" />
</LinearLayout >
```

```
public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void clicked(View view) {
        EditText input = (EditText) findViewById(R.id.input);
        TextView output = (TextView) findViewById(R.id.output);
        int num = Integer.parseInt(input.getText().toString());

        if (num > 90)
            output.setText("A+");
        else if (num >= 80)
            output.setText("A");
        else if (num >= 70)
            output.setText("B");
        else if (num >= 60)
            output.setText("C");
        else if (num >= 50)
            output.setText("D");
        else
            output.setText("F");
    }
}
```

14. Code the XML for this Button. Then, create the corresponding method in Java.

/5

Reverse

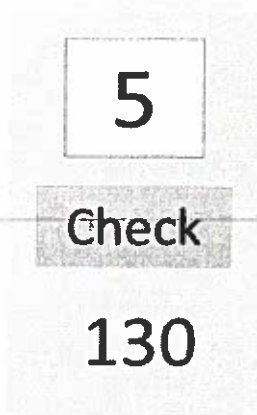
50 sp text size
40 dp padding
onClick: backwards
id: rev

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="50sp"  
    android:padding="40dp"  
    android:onClick="backwards"  
    android:id="@+id/rev"  
    android:text="Reverse"  
>
```

```
public void backwards (View view) { }
```

15. A number of test cases were run on this app and the results of each test are shown in the table below.
What is the code in the onClick method named check? (The editText's id is input; the textView's is output)

/5



```
public void check (View view) {  
    EditText input = (EditText) findViewById (R.id . input);  
    TextView output = (TextView) findViewById (R.id . output);  
    int n = Integer.parseInt (input.getText (). toString ());  
    int ans = n * n * n + n;  
    output.setText (ans + " ");  
}
```

Test Case	Output
4	68
1	2
120	1728120
0	0
-135	-2460510
2	10