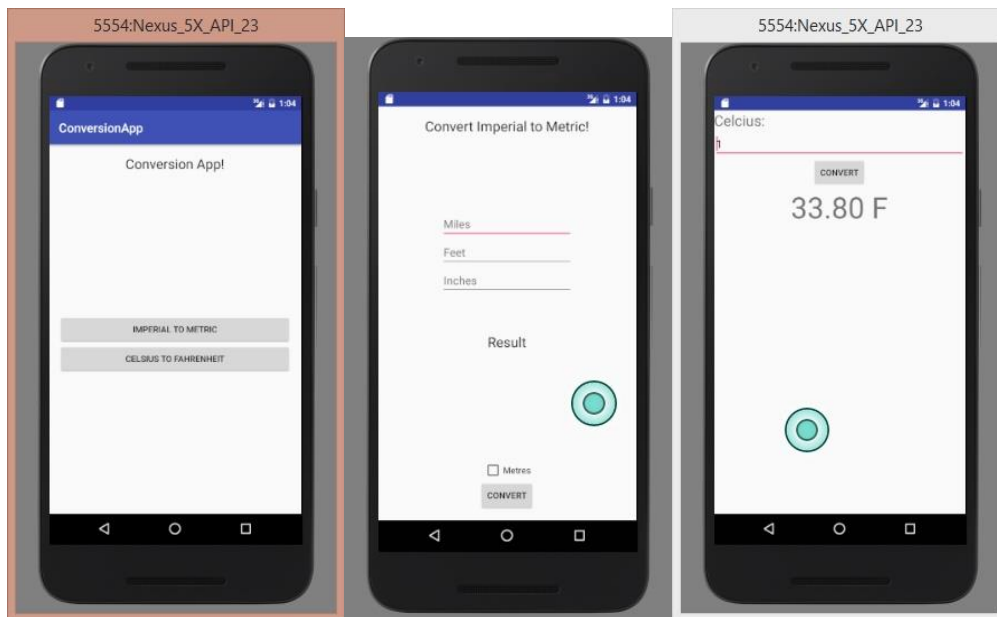
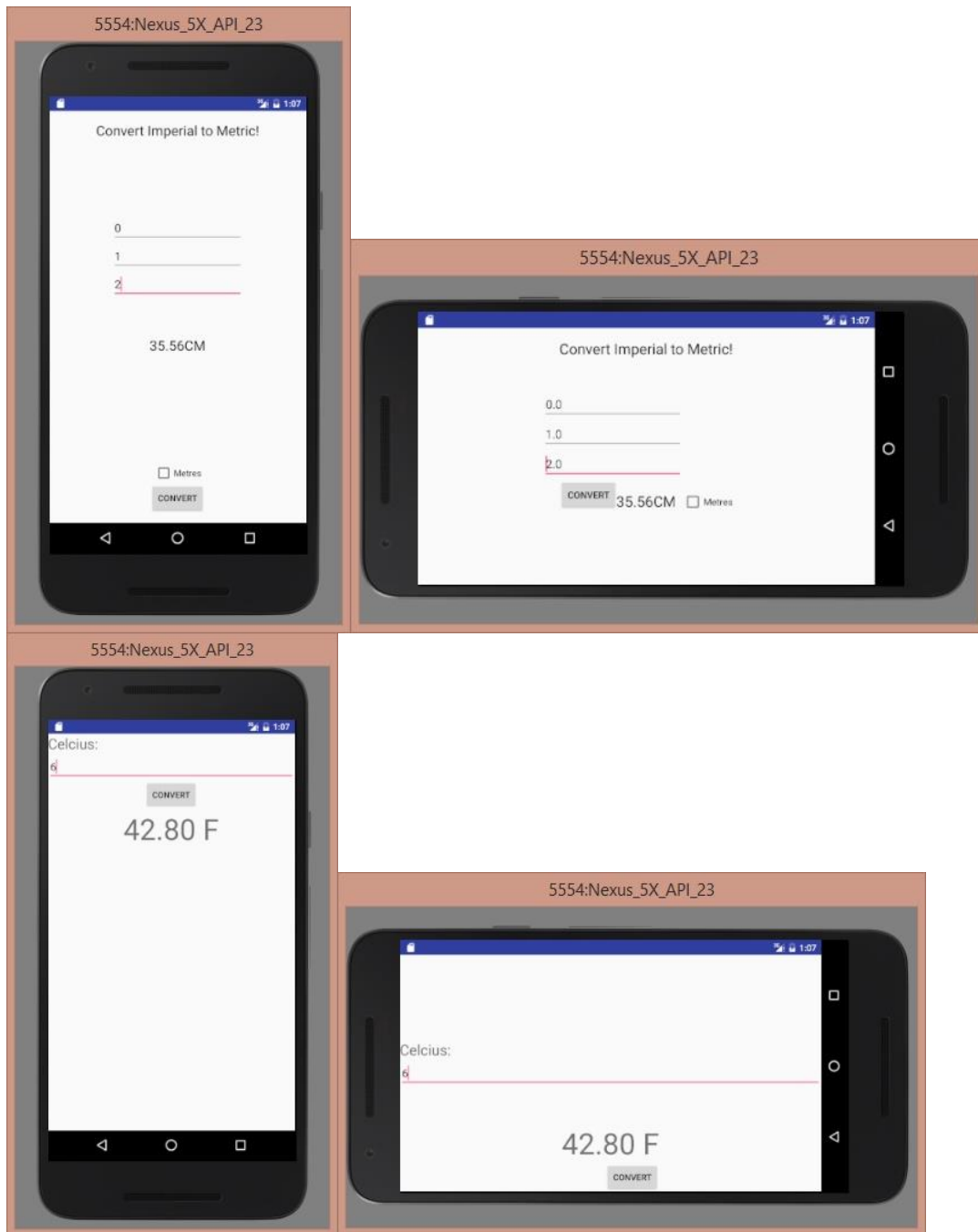


Task 1

- App has a total of three different activities.



- Properly support orientation changes. Specifically, it should not lose values/state information.



- Provide code snippets showing how different activities are connected.

```

Button b1=(Button)findViewById(R.id.t);
Button b2=(Button)findViewById(R.id.m);
//on click send user to other activities with intent
b1.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        Intent myintent2 = new Intent(MainActivity.this, TemperatureConverter.class);
        startActivity(myintent2);
    }

});
b2.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        Intent myintent2 = new Intent(MainActivity.this, MeasurementConverter.class);
        startActivity(myintent2);
    }

});

```

- Provide code snippets showing how orientation change is managed.

```

//saving values for new instance when rotated
@Override
protected void onSaveInstanceState(Bundle dataBundle) {
    super.onSaveInstanceState(dataBundle);

    dataBundle.putDouble("Celsius", c);
    dataBundle.putDouble("Fahrenheit", f);
}

//getting data from when the instance is started again
@Override
protected void onRestoreInstanceState(Bundle dataBundle) {
    super.onRestoreInstanceState(dataBundle);

    c = dataBundle.getDouble("Celsius");
    f = dataBundle.getDouble("Fahrenheit");
}

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_temperature_converter);
    //checking to see if any saves for data from previous instances
    if(savedInstanceState != null)
    {
        EditText inputTempText = (EditText) findViewById(R.id.inputTempEditText);
        TextView convertedText = (TextView) findViewById(R.id.convertedTempTextView);
        convertedText.setText(String.valueOf(f)+" F");
        inputTempText.setText(String.valueOf(c));
    }
    initializeUI();
}

```

- Provide code snippet showing the XML layouts of the activities.

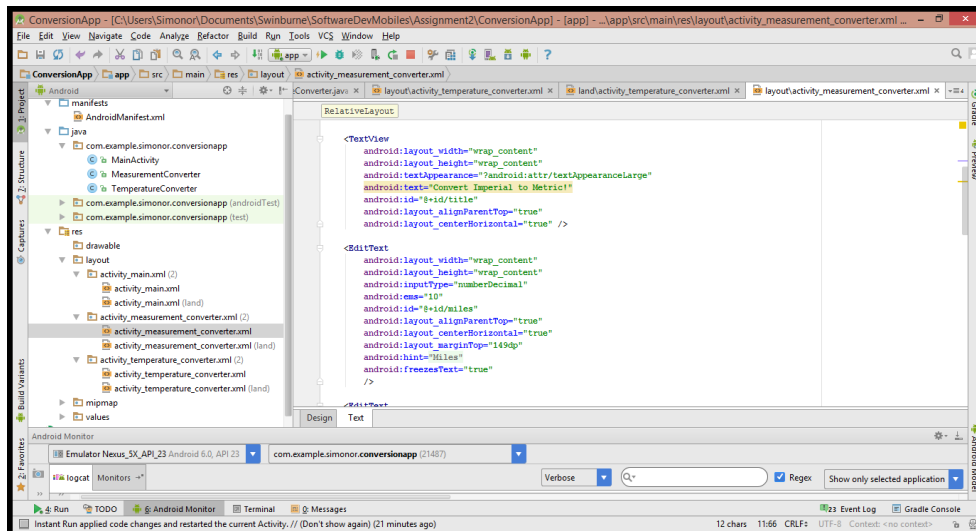
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:id="@+id/celciusTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:text="Celcius:"
    />

    <EditText android:id="@+id/inputTempEditText"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="numberSigned|numberDecimal"
        android:text="1"
        android:freezesText="true"
    />

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.co
    android:screenOrientation="landscape"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView android:id="@+id/celciusTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:text="Celcius:"
        android:layout_above="@+id/inputTempEditText"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <EditText android:id="@+id/inputTempEditText"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="numberSigned|numberDecimal"
        android:text="1"
        android:freezesText="true"
```



- Code snippets are commented appropriately

```
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_temperature_converter);
    //checking to see if any saves for data from previous instances
    if(savedInstanceState != null)
    {
        EditText inputTempText = (EditText) findViewById(R.id.inputTempEditText);
        TextView convertedText = (TextView) findViewById(R.id.convertedTempTextView);
        convertedText.setText(String.valueOf(f) + " F");
        inputTempText.setText(String.valueOf(c));
    }
    initializeUI();
}

//setup ui
private void initializeUI()
{
    Button convertButton = (Button)findViewById(R.id.convertButton);
    //getting all variables for text boxes, check boxes and edit text boxes
    public void Initialize()
    {
        mSet = (CheckBox) findViewById(R.id.checkBox);
        Tmiles = (EditText)findViewById(R.id.miles);
        Tfeet = (EditText)findViewById(R.id.feet);
        T inches = (EditText)findViewById(R.id.inches);
        results = (TextView)findViewById(R.id.result);
        //setting up button for converting
        Button convertButton = (Button)findViewById(R.id.convertButton);
        convertButton.setOnClickListener(convertBtnListener);
    }
    //when clicked do this
    private View.OnClickListener convertBtnListener = new View.OnClickListener()
    {
        public void onClick(View v)
        {
            setResults();
        }
    };
    //assigning the result and setting the text box with the value
    public void setResults()
    {
        measurementCheck();

        setImperial();

        results.setText(String.valueOf(result) + measurement);
    }
}
```

Task 2

- Code snippets show how the two activities are connected.

```
//saving choice of picture into string then loading other activity
public void sendIntent(View V)
{
    Intent intent = new Intent(this, ImageDisplay.class);
    ImageButton clicked = (ImageButton) findViewById(V.getId());
    String message = clicked.getContentDescription().toString();
    intent.putExtra(EXTRA_MESSAGE, message);
    startActivity(intent);
}

//getting information sent by other activity
Intent intent = getIntent();
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
```

- Code snippet shows the XML layouts of the activities.

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">
    <TableRow
        android:id="@+id/tableRow1"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:gravity="center">
        <ImageButton
            android:id="@+id/imageView1"
            android:src="@mipmap/katsucurry"
            android:adjustViewBounds="false"
            android:contentDescription="@string/katsucurry"/>
        <ImageButton
            android:id="@+id/imageView2"
            android:src="@mipmap/pho"
            android:contentDescription="@string/pho"/>
    </TableRow>
</TableLayout>
```

```
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:contentDescription="@string/user_selected" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Medium Text"
    android:id="@+id/textView"
    android:layout_alignTop="@+id/imageView"
    android:layout_centerHorizontal="true" />
```

- Code snippets are commented appropriately.

```
public class ImageDisplay extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_image_display);
        //getting information sent by other activity
        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        setPicture(message);
    }

    public void setPicture(String message)
    {
        //converting message to include drawable to be located in drawable folder
        message = "@drawable/" + message;
        //getting resource ID
        int resID = getResources().getIdentifier(message, "drawable", getPackageName());
        //getting imageView to put picture into
        ImageView img = (ImageView) findViewById(R.id.imageView);
        TextView txt = (TextView) findViewById(R.id.textView);
        txt.setText(message);
        //assigning picture to imageView
        img.setImageResource(resID);
    }
}
```



```

public class MainActivity extends AppCompatActivity {
    public final static String EXTRA_MESSAGE = "AAA";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //finding buttons in layout
        ImageButton b1 = (ImageButton) findViewById(R.id.imageView1);
        ImageButton b2 = (ImageButton) findViewById(R.id.imageView2);
        ImageButton b3 = (ImageButton) findViewById(R.id.imageView3);
        ImageButton b4 = (ImageButton) findViewById(R.id.imageView4);
        //setting listener to activate when clicked
        b1.setOnClickListener(onClickListener);
        b2.setOnClickListener(onClickListener);
        b3.setOnClickListener(onClickListener);
        b4.setOnClickListener(onClickListener);
    }
    //on click of any button send to other layout
    private View.OnClickListener onClickListener = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            sendIntent(v);
        }
    }
}

```

- Code snippet shows how the image and text are loaded/set

```

public void setPicture(String message)
{
    //converting message to include drawable to be located in drawable folder
    message = "@drawable/" + message;
    //getting resource ID
    int resID = getResources().getIdentifier(message, "drawable", getPackageName());
    //getting imageView to put picture into
    ImageView img = (ImageView) findViewById(R.id.imageView);
    TextView txt = (TextView) findViewById(R.id.textView);
    txt.setText(message);
    //assigning picture to imageView
    img.setImageResource(resID);
}

```

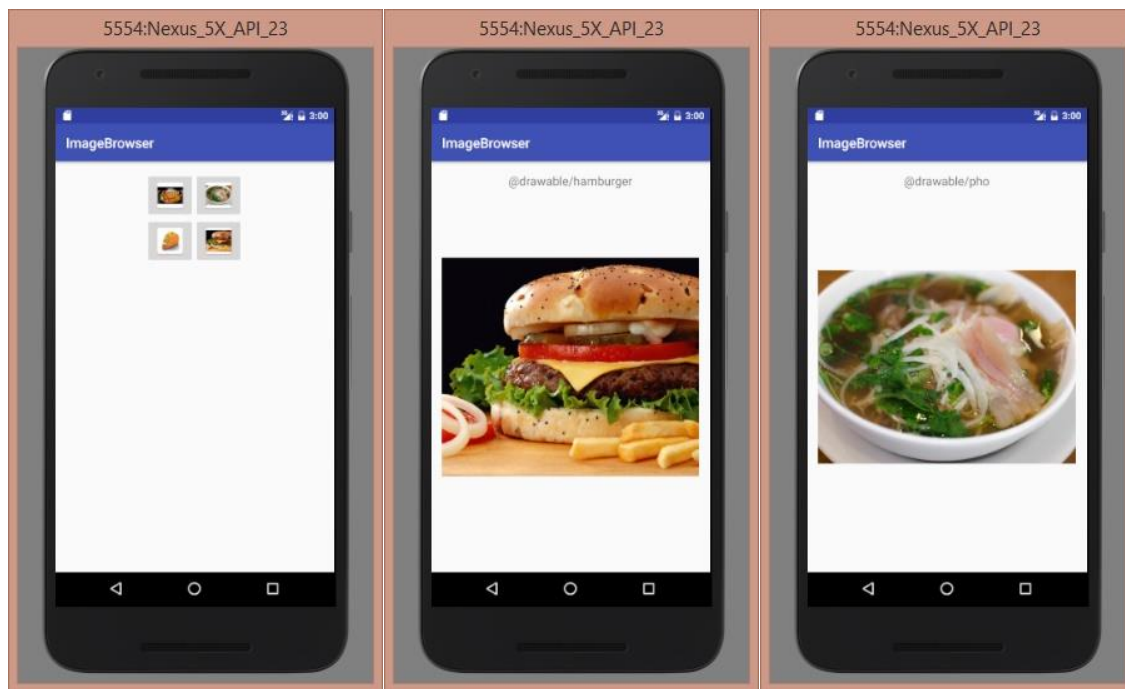
- String values are in external XML values (not within layout)

```

<resources>
    <string name="app_name">ImageBrowser</string>
    <string name="katsucurry">katsucurry</string>
    <string name="pho">pho</string>
    <string name="taco">taco</string>
    <string name="hamburger">hamburger</string>
    <string name="user_selected">User selected</string>
</resources>

```

- Screen shots show how the app works



Task 3

(a) Why is the intent messaging facility considered as a late run-time binding between components?

Late runtime binding means that the compiler does not check intent code so that if there are errors the app will fail. Intents are just information packets so there is no reason why the compiler should check a message between components during runtime. It does not have a compile time binding between components

(b) What are the contents of the passive data structure (of an intent)?

The content of an intent is holding an abstract description of an action to be performed and holding some data in a bundle.

(c) Why is the word "passive" used for the intent data structure? (Hint: What is the responsibility of an Intent object? Do they have any inherent intelligence built into them?)

The intent system is completely asynchronous. The receiver will get the message only if they request to see intents. Intents do not perform the action themselves, instead they redirect the description for another activity perform it.

(d) Use an example to better explain the sentence "abstract description of an operation to

be performed”.

```
Intent email = new Intent(Intent.ACTION_SEND, Uri.parse("mailto:));
```

```
startActivity(Intent.createChooser(email, "Choose an email client from..."));
```

This Intent is an abstract description, in which the description is ordering any activity that can respond for example, writing an email. The intent is redirecting the operation to be performed to an email program with the <mailto:> command. On a device certain activities will respond to this call of intent and use the description to then perform an action such as writing an email.