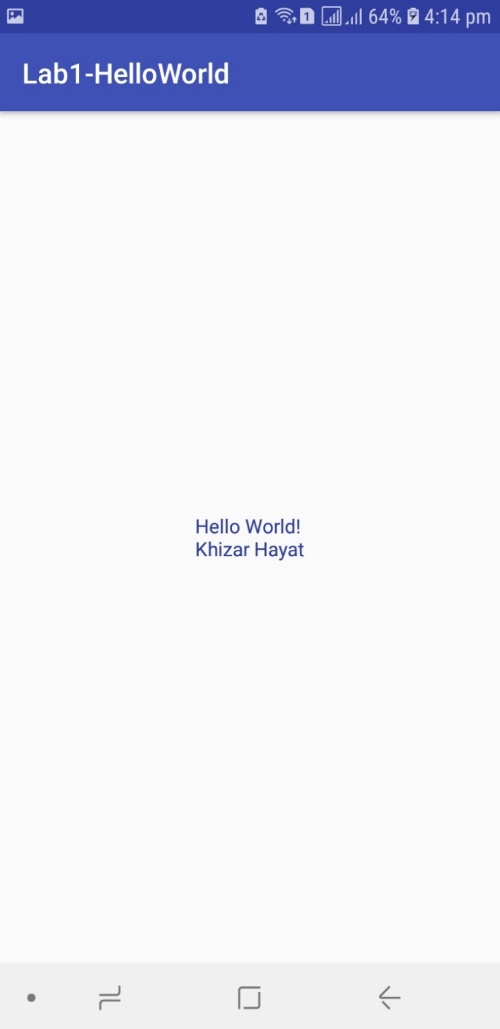
**Task#1:** Create Simple Hello World! App Run on android virtual device and on your android phone.

**Solution:**

**XML:**

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:textColor="@color/colorPrimaryDark"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</android.support.constraint.ConstraintLayout>

**Output:**



**Task#2:** Why we need separate folder for Assets and Resources if both folder are for same purpose?

**Solution:**

**Assets:**

This is empty. You can use it to store raw asset files. Files that you save here are compiled into an .apk file as-is, and the original filename is preserved. You can navigate this directory in the same way as a typical file system using URIs and read files as a stream of bytes using the AssetManager. For example, this is a good location for textures and game data.

**Res:**

For arbitrary raw asset files. Saving asset files here instead of in the assets/ directory only differs in the way that you access them. These files are processed by aapt and must be referenced from the application using a resource identifier in the R class. For example, this is a good place for media, such as MP3 or Ogg files.

Both are pretty similar. The real main difference between the two is that in the res directory each file is given a pre-compiled ID which can be accessed easily through R.id.[res id]. This is useful to quickly and easily access images, sounds, icons...

The assets directory is more like a filesystem and provides more freedom to put any file you would like in there. You then can access each of the files in that system as you would when accessing any file in any file system through Java. This directory is good for things such as game details, dictionaries etc.

**Task#3:** Explain the directory structure of simple Hello World app created in Task#1.

**Solution:**

Every Android project contains several folders:

* *src*: This folder contains the Java source files.
* *gen*: Generated Java library, this library is for Android internal use only.
* *Res*: Here we can store resource files such as pictures, XML files for defining layouts, and so forth. Within this folder there are additional folders such as *Drawable*, *Layout*, and *Values*.
* *Drawable*: Here we store the various graphic files. We can see three types of drawable folders. This is because there are many Android devices with different screen resolutions. By default, there are several versions of this folder such as: *Drawable-mdpi*, *drawable-hdpi*, and so forth. This is required in order to adapt to different screen resolutions.
* *Layout*: This is the place for XML layout files. Layout files are XML files which define how various Android objects (such as textboxes, buttons, etc.) are organized on the screen.
* *Values*: XML files which store various string values (titles, labels, etc.).

Major files in the Android project:

* *AndroidManifest.xml*: This is the Android definition file. It contains information about the Android application such as minimum Android version, permission to access Android device capabilities such as internet access permission, ability to use phone permission, etc.
* *MainLayout.xml*: This file describes the layout of the page. This means the placement of every component (such as textboxes, labels, radio buttons, user defined components, etc.) on the app screen.
* Activity class: Every application that occupies the entire device screen needs at least one class which inherits from the Activity class. One major method is called OnCreate. This method initiates the app and loads the layout page.