**Experiment 03: Write program to implement frame layout, table layout and relative layout**

**Learning Objective:** To write a program to implement frame layout, table layout and relative layout.

**Tools:** Android Studio

**Theory:**

Android Studio:

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. To support application development within the Android operating system, Android Studio uses a Gradle- based build system, emulator, code templates, and GitHub integration. It is an open source software platform and operating system for mobile devices which is based on the Linux kernel. It was developed by Google and later by the Open Handset Alliance (OHA). It allows writing managed code in the Java language and Kotlin language. It has its own virtual machine i.e. DVM (Dalvik Virtual Machine), which is used for executing the android application.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these call-backs as an activity enters a new state. As the user begins to leave the activity, the system calls methods to dismantle the activity.

**Table Layout:**

It is a layout that consists of rows and columns. It is further divided as [TableRow](https://developer.android.com/reference/android/widget/TableRow) objects, each defining a row a layout. A layout that arranges its children horizontally. A TableRow should always be used as a child of a [TableLayout.](https://developer.android.com/reference/android/widget/TableLayout) If a TableRow's parent is not a TableLayout, the TableRow will behave as a horizontal [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout). The children of a TableRow do not need to specify the layout\_width and layout\_height attributes in the XML file. TableRow always enforces those values to be respectively [View Group.Layout Params](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams#MATCH_PARENT)

[.MATCH\_PARENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams#MATCH_PARENT) and [View Group . Layout Params. WRAP\_CONTENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams#WRAP_CONTENT). TableLayout containers do not display border lines for their rows, columns, or cells. Each row has zero or more cells; each cell can hold one [View](https://developer.android.com/reference/android/view/View) object. The table has as many columns as the row with the most cells. A table can leave cells empty. Cells can span columns, as they can in HTML.

**Frame Layout:**

It is one of the simplest layout to organize view controls. Frame Layout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. We can add multiple children to a FrameLayout and control their position by assigning gravity to each child, using the android:layout\_gravity attribute.

**Relative Layout:**

RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center). It is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested LinearLayout groups, you may be able to replace them with a single RelativeLayout. The Relative Layout is very flexible layout used in android for custom layout designing. It gives us the flexibility to position our component/view based on the relative or sibling component’s position. Just because it allows us to position the component anywhere we want so it is considered as most flexible layout.

**Implementation:**

**Output:**

**Result and Discussion:** After performing the experiment I was able to implement frame layout, table layout and relative layout.

**Learning Outcomes:** The student should have the ability to

LO1: **explain** the details of Android Virtual Device.

LO2: **execute** a simple program on Android Studio

**Course Outcomes:** Upon completion of the course students will be able to execute a simple program on Android Studio.

**Conclusion:** We understood in this experiment the concept of basic application layout used in the android studios such as Table layout, Frame layout and Relative layout. We implemented the fundamental program having frame layout, table layout and relative layout and displayed the output on the Android Virtual Device

For Faculty Use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| **Marks Obtained** |  |  |  |