Assignment:

Name: **Ateeq ullah**
Reg: **fa18-bse-050**
Subject: **Android Application development**

==========

Difference between Constraintlayout and Relativelayout

Constraint Layout has dual power of both Relative Layout as well as Linear layout: Set relative positions of views (like Relative layout) and also set weights for dynamic UI (which was only possible in Linear Layout).
A very powerful use is grouping of elements by forming a chain. This way we can form a group of views which as a whole can be placed in a desired way without adding another layer of hierarchy just to form another group of views.
In addition to weights, we can apply horizontal and vertical bias which is nothing but the percentage of displacement from the centre. (bias of 0.5 means centrally aligned. Any value less or more means corresponding movement in the respective direction) .
Provides power of automatic constraint applying by the use of Blue print and Visual Editor tool which makes it easy to design a page.
I've measured and compared the time of onCreate method and time between a start of onCreate and end of execution of last preformDraw method which visible in CPU monitor. All test were done on Samsung S5 mini with android 6.0.1 Here results:
Fresh start (first screen opening after application launch)
Relative Layout
OnCreate: 123ms
Last preformDraw time - OnCreate time: 311.3ms
Constraint Layout
OnCreate: 120.3ms
Last preformDraw time - OnCreate time: 310ms
Besides that, found that on loop counts less than 100 constraint layout variant is faster during execution of inflating, measure, and layout then variants with Relative Layout. And on old Android devices, like Samsung S3 with Android 4.3, the difference is bigger.

(1) Relative Layout:

android:layout_centerInParent="true"

(1) Constraint Layout equivalent :

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintStart_toStartOf="parent"
app:layout_constraintRight_toRightOf="parent" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent"

---

(2) Relative Layout:

android:layout_centerHorizontal="true"

(2) Constraint Layout equivalent:

app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintStart_toStartOf="parent"
app:layout_constraintRight_toRightOf="parent" app:layout_constraintEnd_toEndOf="parent"

---

(3) Relative Layout:

android:layout_centerVertical="true"

(3) Constraint Layout equivalent:

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintTop_toTopOf="parent"

---

(4) Relative Layout:

android:layout_alignParentLeft="true"

(4) Constraint Layout equivalent:

app:layout_constraintLeft_toLeftOf="parent"

---

(5) Relative Layout:

android:layout_alignParentStart="true"

(5) Constraint Layout equivalent:

app:layout_constraintStart_toStartOf="parent"

---

(6) Relative Layout:

android:layout_alignParentRight="true"

(6) Constraint Layout equivalent:

app:layout_constraintRight_toRightOf="parent"

---

(7) Relative Layout:

android:layout_alignParentEnd="true"

(7) Constraint Layout equivalent:

app:layout_constraintEnd_toEndOf="parent"

---

(8) Relative Layout:

android:layout_alignParentTop="true"

(8) Constraint Layout equivalent:

app:layout_constraintTop_toTopOf="parent"

---

(9) Relative Layout:

android:layout_alignParentBottom="true"

(9) Constraint Layout equivalent:

app:layout_constraintBottom_toBottomOf="parent"

---

(10) Relative Layout:

android:layout_alignStart="@id/view"

(10) Constraint Layout equivalent:

app:layout_constraintStart_toStartOf="@id/view"

==========