UNIT-5 : Fragments :-

⊛ A fragment is a piece of an activity which enable more modular activity design. It can be called as sub-activity.

⊛ Fragment has its own Layout and its own behaviour with its own lifecycle call backs.

⊛ You can remove or add fragments in an activity while the activity is running.

⊛ You can combine multiple fragments in a single activity to build a multi-pane UI.

⊛ A fragment can be used in multiple activities.

⊛ Fragment's lifecycle is closely related to the lifecycle of its host activity.

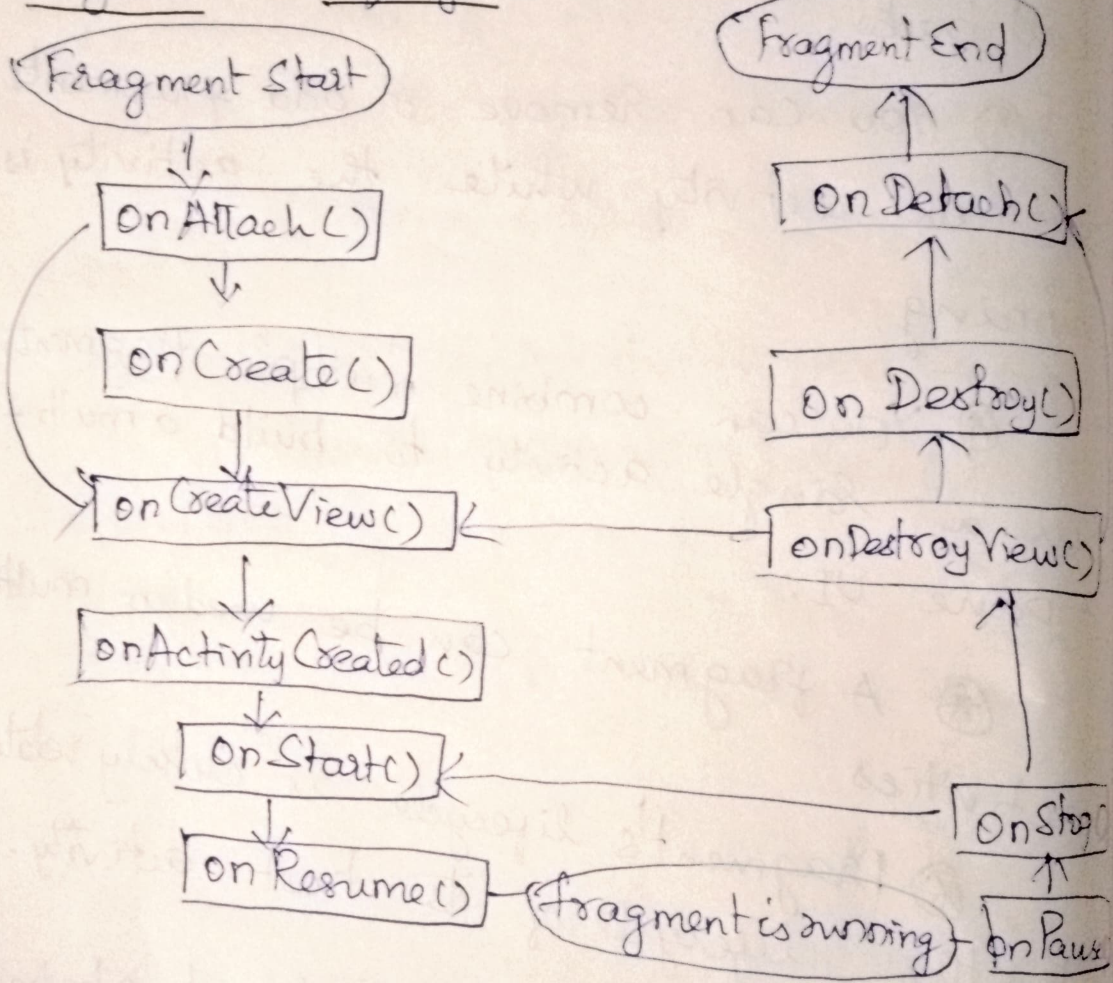
⊛ A fragment can implement a behavior that has no user interface components.

⊛ Fragments are added in Honeycomb version of Android API Version 11.

How it is created?-

- By extending the Fragment class in java file.
- In xml file by adding <Fragment> element

Fragment's Life cycle:-



on Attach (Activity) — It is called once when it is attached with activi

on Create (Bundle) — It is used to initializin the fragment

on CreateView (Layout Inflater, ViewGroup, Bundle) — Creates & returns View hierarchy onActivityCreated(Bundle) - It is invoked after the completion of onCreate() method.

onStart() — makes the fragment visible

on Resume() — makes the fragment interactive

onPause () - is called when fragment is no longer ~~visible~~ interactive

onStop() — is called when fragment is no longer visible

onDestroyView() — allows the fragment to Clean up resources.

onDestroy() - allows the fragment to do final clean up of fragment state

onDetach() - it is called immediately prior to the fragment no longer being associated with its activity.

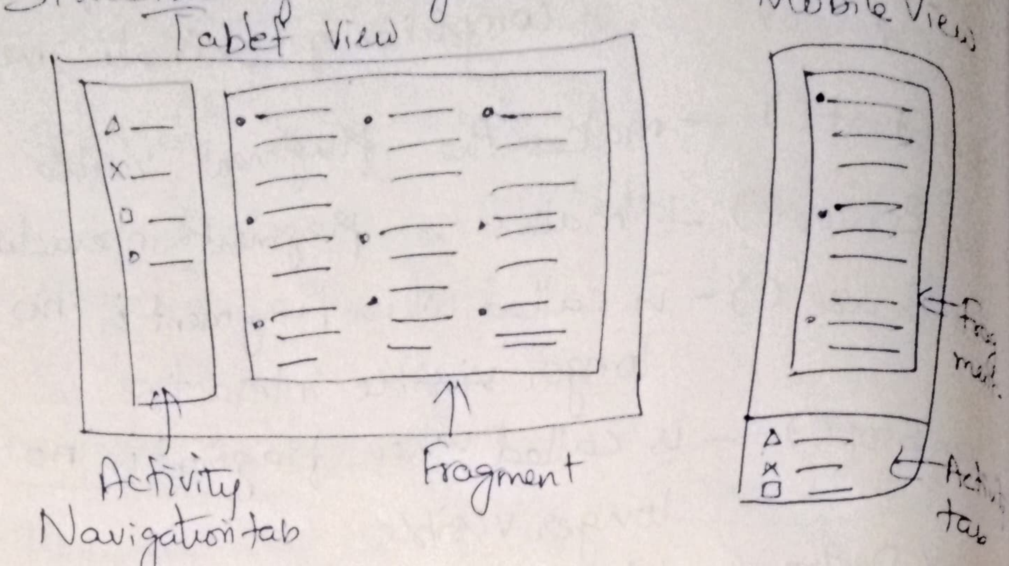
<u>Steps to Use Fragments</u>:-

* First decide how many fragments you want to use in activity

* Based on number of fragments, create classes which will extend the Fragment class

* Corresponding to each fragment, create layout files in XML files These files will have layout for the defined fragments.

Structure of Fragment :-



Displaying & Fetching Information Using Dialogs & Fragments:

Dialog :- A dialog is a small window that pops up to interact with the user. It can display important messages and can even prompt for some data Once the interaction with the dialog is over, the dialog disappears, allowing the user to continue with the application

Fragment :- As the name suggests, enable us to fragment or divide our Activities into encapsulated reusable modules, each with its own user interface, making our application suitable to different screen sizes.

When is Dialog used?

We usually create a new activity or screen for interacting with users, but when we want only a little information or want to display an essential message, dialogs are preferred. Dialogs are also used to guide users in providing requested information, confirming certain actions, and displaying warnings or error messages. The following is an outline of different dialog window types provided by the Android SDK.

(*) Dialog - The basic class for all dialog types

(*) AlertDialog - A dialog with one, two or three Button controls.

(*) CharacterPickerDialog - A dialog that enables you to select an accented characters associated with a regular character source.

(*) DatePickerDialog :- A dialog that enables you to set and select a date with a DatePicker Control.

⊛ ProgressDialog – A dialog that displays a ProgressBar Control

⊛ TimePickerDialog – A dialog that enables you to set & Select a time with a TimePicker control.

How to Create a Dialog?–

A dialog is created by creating an instance of the Dialog class. The Dialog class creates a dialog in the form of a floating window containing messages and controls for user interaction. In Android the dialogs are called asynchronously.

Each dialog window is defined within the activity where it will be used. A dialog window can be created once and displayed several times. It can also be updated dynamically.

The following is a list of the Activity class dialog methods:

showDialog (), onCreateDialog(), onPrepareDialog(), dismissDialog(), removeDialog()

All these methods are deprecated with the new preferred way being to use the DialogFragment with the FragmentManager.

Selecting Date & Time in One Application:-