Handlers

Handlers

A Handler allows you to send and process Message and Runnable objects associated with a thread's MessageQueue

- **Post()** – it going to post message from background thread to main thread using looper.
- **sendmessage()** – if you want to organize what you have sent to ui (message from background thread) or ui functions. you should use sendMessage().

postDelayed(Runnable r, Object token, long delayMillis) **Causes the Runnable r to be added to the message queue, to be run after the specified amount of time elapses**. final void.

removeCallbacks(Runnable r) Remove any pending posts of Runnable r that are in the message queue

++i increments the value, then returns it.

i++ returns the value, and then increments it.

Service : is a component of android which performs long running operation in background, mostly with out having UI.

Thread : is a O.S level feature that allow you to do some operation in the background.

Though conceptually both looks similar there are some crucial differentiation.

1.Service - if it is destroyed while performing its job, in the middle by Android due to low memory scenario. Then android will make sure that it will restart your service.

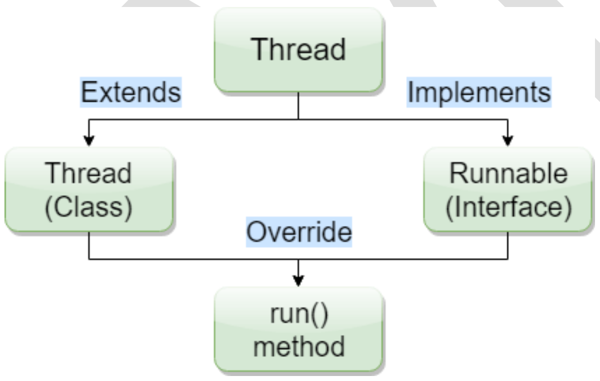
2.Thread - if it is destroyed by android in middle due to low memory, then android will not guarantee to restart it again. That means user lost his half work.

3.Service - is a component of android, so it has priority levels to be considered while destroying an application due to low memory.

4. Thread - is not a component of android, so android will not take thread priority into consideration while killing an application due to low memory.

The Runnable interface should be implemented by any class whose instances are intended to be executed by a thread. The class must define a method of no arguments called run.

This interface is designed to provide a common protocol for objects that wish to execute code while they are active. For example, Runnable is implemented by class Thread. Being active simply means that a thread has been started and has not yet been stopped.



ContentValues are used to insert new rows into tables. Each Content Values object represents a single table row as a map of column names to values.

Queries in Android are returned as Cursor objects. Rather than extracting and returning a copy of the result values, Cursors are pointers to the result set within the underlying data. Cursors provide a managed way of controlling your position (row) in the result set of a database query.

The Cursor class includes a number of navigation functions including, but not limited to, the following:

> moveToFirst Moves the cursor to the first row in the query result

> moveToNext Moves the cursor to the next row

> moveToPrevious Moves the cursor to the previous row

> getCount Returns the number of rows in the result set

> getColumnIndexOrThrow Returns the index for the column with the specified name (throwing an exception if no column exists with that name)

> getColumnName Returns the name of the specified column index

> getColumnNames Returns a string array of all the column names in the current Cursor

> moveToPosition Moves the Cursor to the specified row

> getPosition Returns the current Cursor position

Android provides a convenient mechanism for simplifying the management of Cursors within your Activities. The startManagingCursor method integrates the Cursor's lifetime into the calling Activity's. When you've finished with the Cursor, call stopManagingCursor to do just that.

******