

## ❖ Android AsyncTask

- ✓ Android AsyncTask is an abstract class provided by Android which gives us the liberty to perform heavy tasks in the background and keep the UI thread light thus making the application more responsive.
  - ✓ Android application runs on a single thread when launched. Due to this single thread model tasks that take longer time to fetch the response can make the application non-responsive.
  - ✓ To avoid this we use android AsyncTask to perform the heavy tasks in background on a dedicated thread and passing the results back to the UI thread.
  - ✓ Hence use of AsyncTask in android application keeps the UI thread responsive at all times.
- The basic methods used in an android AsyncTask class are defined below :
- ✓ **doInBackground()** : This method contains the code which needs to be executed in background. In this method we can send results multiple times to the UI thread by **publishProgress()** method. To notify that the background processing has been completed we just need to use the return statements
  - ✓ **onPreExecute()** : This method contains the code which is executed before the background processing starts
  - ✓ **onPostExecute()** : This method is called after doInBackground method completes processing. Result from doInBackground is passed to this method
  - ✓ **onProgressUpdate()** : This method receives progress updates from doInBackground method, which is published via publishProgress method, and this method can use this progress update to update the UI thread
- The three generic types used in an android AsyncTask class are given below :
- ✓ **Params** : The type of the parameters sent to the task upon execution
  - ✓ **Progress** : The type of the progress units published during the background computation
  - ✓ **Result** : The type of the result of the background computation

## ❖ Android AsyncTask Example

- To start an AsyncTask the following snippet must be present in the MainActivity class :

```
MyTask myTask = new MyTask();  
  
myTask.execute();
```

- ✓ MyTask is a sample class, it extends from AsyncTask & execute method is used to start the background thread.

## ❖ Note :

- The AsyncTask instance must be created and invoked in the UI thread.
- ✓ The methods overridden in the AsyncTask class should never be called.
  - ✓ They're called automatically (**Callback methods**)
  - ✓ AsyncTask can be called only once. Executing it again will throw an exception