Asynchronous Apex:

Day 1:

* Annotate with @future for the method to run in asynchronous mode
* Cannot call future method from another future method
* Use **system.isFuture()** to check the method if it is called from another future method
* Maximum of 2,50,000 Async jobs can be processed.
* Maximum of 50 jobs can be processed in single execution.
* To check limits:
  + Limits.getFutureCalls()
  + Limits.getLimitFutureCalls()

Day 2:

* To add endpoints in the SF Org then Settings🡪Home🡪Remote site settings and add the URL
* How u will annotate external endpoint in future method use **@future(callout=true)**
* To integrate with endpoint:
  + Http – to send request
  + Httprequest – to define end point
  + Httpresponse – to receive response
* To get statuscode🡪response.getstatuscode()==200(Success)
* To get body🡪response.getbody()

Day 3:

* When an user account is created without role then there wont be mixed DML Exception
* For Mixed DML Exception use @future annotation
* To pass parameter from trigger to future method use serialize in the trigger and convert into string and then pass it as a parameter
* Then this string parameter in the future method can be deserialized and converted back in to list
* To check status in **AsyncApexJob:**
  + **List<AsyncApexJob> lst=[Select Id,Status from AsyncApexJob]**
  + Delete in the AsyncApexjob cannot be used
  + Instead use
    - Integer Count=System.purgeoldAyncJobs(Date.today()+1)
    - Will delete the jobs as mentioned before day of the date.
  + Query:Status,Queued,Preparing,Processessing,Aborted,Failed,Completed

Day 4:

* Queueable Apex
* Use implements queueable in class and in execute method use queueablecontext ctx
* To get job id then use ctx.getJobID
* To run the job use **System.enqueueJob(new methodname());**
* Since execute method cannot receive parameter use constructor to receive parameter from triggers
* Chaining Jobs:
  + One parent method can have only one child method
  + Chaining cannot be considered in test method so use **test.isrunning()** in a child record

Day 5:

* Transaction Finalizer:
  + If Queueable job fails then using System.Finalizer makes the job to run irrespective of other queueable jobs
  + Failed jobs can be reenqueued 5 times

Day 6:

* Batchable Apex:Proceess record in smaller chunks
* Process about 50 Million records
* Database.QueryLocator
* Default size-200

Day 8:

* We cannot pass variable from one method to another method
* Use Database.stateful to assign the variable