**21.2)**

**1. Explain the different methods of ● mapper class and ● reducer class in brief.**

The three main methods in the mapper class are a.set up method b.map method c.clean up method

The three main methods in the reducer class are a.set up method b.reduce method c.clean up method

1.Map method in mapper Map is the most prominent method of the Mapper class. The syntax is defined below −

map(KEYIN key, VALUEIN value, org.apache.hadoop.mapreduce.Mapper.Context context)

This method is called once for each key-value pair in the input split. Maps a single input key/value pair into an intermediate key/value pair. Output pairs need not be of the same types as input pairs. A given input pair may map to zero or many output pairs.

**2. Reduce method in reducer Reduce is the most prominent method of the Reducer class. The syntax is defined below −**

Reduce(KEYIN key, Iterable values, org.apache.hadoop.mapreduce.Reducer.Context context)

This method is called once for each key on the collection of key-value pairs. Each reduce function processes the intermediate values for a particular key generated by the map function and generates the output. Essentially there exists a one-one mapping between keys and reducers. Several reducers can run in parallel, since they are independent of one another. The number of reducers is decided by the user. By default, the number of reducers is 1.

**3.Set up and clean up methods in mapper and reducer. The lifecycle of a map/reduce task is (from a programmer’s point of view):**

setup -> map -> cleanup

setup -> reduce -> cleanup

**a.setup:**

Called once at the beginning of the task We can use hadoop setup method if you want to define user defined parameters to your map task or reducer task and you want to compare the words with mapreduce input file and if you want to use hadoop distributed cache in your mapreduce program at that time we will hadoop setup method.

**b.cleanup:**

Called once at the end of the task. What typically happens during cleanup() is that you clean up any resources you may have allocated. There are other uses too, which is to flush out any accumulation of aggregate results.

As already mentioned, setup() and cleanup() are methods you can override, if you choose, and they are there for you to initialize and clean up your map/reduce tasks.You actually don’t have access to any data from the input split directly during these phases. The lifecycle of a map/reduce task is (from a programmer’s point of view):

Set up -> map -> cleanup setup -> reduce -> cleanup

So, for each mapreduce first setup() method is called then map()/reduce() method is called and later cleanup() method is called before exiting the task.

For example, in the canonical word count example, let’s say you want to exclude certain words from being counted (e.g. stop words such as “the”, “is”, etc…). When you configure your MapReduce Job, you can pass a list (comma-delimited) of these words as a parameter (key-value pair) into the configuration object. Then in your map code, during setup(), you can acquire the stop words and store them in some global variable (global variable to the map task) and exclude counting these words during your map logic.