1. Note

- As I am not able to implement the code using CallBack for Daemon to notify Launch and I also observe that when invoking an RMI method, Launch (client) will wait till that method finish executing on the remote server before running next instruction. Hence, I think there is no need of using *completed()* method in *CallBack* class. This led to the way I implemented the *Launch* and *DaemonImpl* class with description below.

2. Description of classes in project

- **Split**: Split takes an input as name of text file to split and default file is hello.txt file in same directory if file name input is missing. Text file is splitting in to two and sending to two daemons via TCP socket and split is client.
- **Launch**: Launch has two threads. One is for invoke RMI calls in Daemons and the other is for running the TCP server socket to receive result files from Daemons. Launch invokes RMI method *rmi_trigger_download()* right after RMI to *call()* method to ask Daemons to transfer result file back to Launch immediately after Daemon done the Map. When the process received all result files, it call the RMI *rmi_daemon_die()* method to ask all Daemons to terminate. After that Launch joins the result files to a file name launch_final_result.txt and executing Reduce and save result to *final_reduce_result.txt*.
- **DaemonImpl**: Daemon receives the block from split via TCP socket and save it to a file (*block#1.txt*), then waiting for Launch to invoke *call()*. *Call()* starts a thread for executing Map on the block file. Another thread executing *rmi_trigger_download()*, it waits till the Map done executing and send the result file name and content of the result file to Launch for Launch to know file from which Daemons. Daemon is client when sending data via TCP socket to Launch. Daemon receives RMI call to *rmi_daemon_die()* method and terminates.
- **Other classes**: fixed typos

3. Instruction to run

- All class files and their compiled class provided in the *Project_code.tar.gz* archive. Extract the tar ball to a destination directory and check if can find all file in *Split* folder.
- Open four terminal tabs or four terminal windows and all change to the *Split* folder.
- Start Daemons first:
    - On two terminal tabs run the command: ***java DaemonImpl 9000*** and **java *DaemonImpl 9001***
- Start Split next (*hello.txt* can be changed with other file put in same directory with *Split* class):
    - On the third terminal tab run the command: ***java Split hello.txt*** or *java Split*
- Start Launch last:
    - Check if in the first two terminals that run the *DaemonImpl* class output *"--> In Main: waiting for Launch to rmi invoke call()*", then on the fourth terminal tab run the command: ***java Launch 9002***