**Algorithm for agent main class:-**

1)      Begin

2)      Start a while loop which will run for 1hrs approx..

3)      Create a seatbelt class’s object and call checkSeatBelt().

4)      Store the returned value of engine and seat from checkSeatBelt() and if seat is 0 and engine is 1 i.e. if seat belt is off and engine is on then violation is committed and print a message accordingly.

5)      Then create an object for wrongDirection class and then call check\_Direction().

6)      Store the returned value of flag and area from check-Direction().

7)      If the flag is 0 then no agent is found in the area. If flag is 1 then it means that the violating agent is found and has been reported successfully. If the flag is something other than 1 or 0 then it means that the agent which was found in the area is going in the same direction as permitted.

8)      Then create an object for speedlimt class and call check\_overspeeding().

9)      Store the returned value of flag1 and area1 from check\_overspeeding().

10)  If flag1 is 0 means that no agent is found in the area. If flag1 is 1 then it means that a violating agent is found and has been reported successfully. If flag1 is something other than 1 or 0 then it means that the agent which was found in the area is going at a speed same as permitted speed.

11)  Then create an object for uninsured\_vehicle class and call getInsuranceDetail().

12)  Store the value of the check returned from getInsuranceDetail().

13)  If the check is 1 then it means that the agent's insurance is up to date and if it is not 1 then it means that agent’s insurance has expired and hasn’t updated even after warning. Thus violation committed which was reported successfully and message was displayed accordingly.

14)  Repeat step 3 to 13 for approx 1 hrs.

15)  End.

**Algorithm for server main class:-**

1)      Begin.

2)      Define port number and socket object.

3)      Create thread pool to execute the code

4)      Start an infinite while loop.

5)      Inside while loop whenever an agent tries to connect to the server then socket connection between server and agent will be created and data can be transferred between server and agent till the connection is closed.

6)      When an agent is connected to the server then the server thread class object is created and executed where it receives agent id and violation type and updates the database.

7)      In server thread class, using sql query fetch violation type and fine to be charged using violation type.

8)      Fetch fine of violated agent from rto table using agent id.

9)      Add both the fine and update rto fine value and set the total fine. RTO ‘s fine column consists of fine which the agents have committed till date.

10)  After that check if a table with name as agent id exists or not. if it exists then don’t create a new table and if it already exists then no need.

11)  Then update the table with violation type, date and time at which violation was committed and fine charged for that violation.

12)  After updating the database, close the connection with the sql database and agent.

13)  Connection between agent and server will be created only when violation is committed and has to be reported by the reporting agent and once the database is updated, connection will be closed.

14)  So everytime agent tries to connect using the server’s port number, repeat steps 5 to 12.

15)  End.