**Ring Election Algorithm Documentation DS Lab 2 (Write up)**

**System Requirements:**

* Eclipse IDE or any other Java IDE
* Windows
* Java

**Compiling and Running the Program:**

1. Import the project into eclipse IDE. Project name is RingApplicationDS.
2. Start the server by running **RelayServer** class as “java application”
3. Start the client by running **Nodes** class as “java application”
   1. The above process pops up a client window
   2. Repeat the same process for as many clients as needed
   3. Node number is mentioned as the window title (incremented for each process spawned)
4. One can then start election process to setup an initial ring network with a coordinator that will be elected by this process.
5. Once the coordinator is elected the token message is passed between the nodes in the ring network. This can be witnessed by looking at the messages that are displayed on each node on the GUI.
6. Now we can crash the coordinator node by clicking on the close button of that process window.
7. It can be seen that the nodes start a re-election if it doesn’t get the connection check token for a particular interval of time (in our case it’s for 50 seconds).
8. All the nodes start the re-election simultaneously and elects a coordinator and the process continues.
9. If a new node is brought up in the network it starts an election and elects a new coordinator.

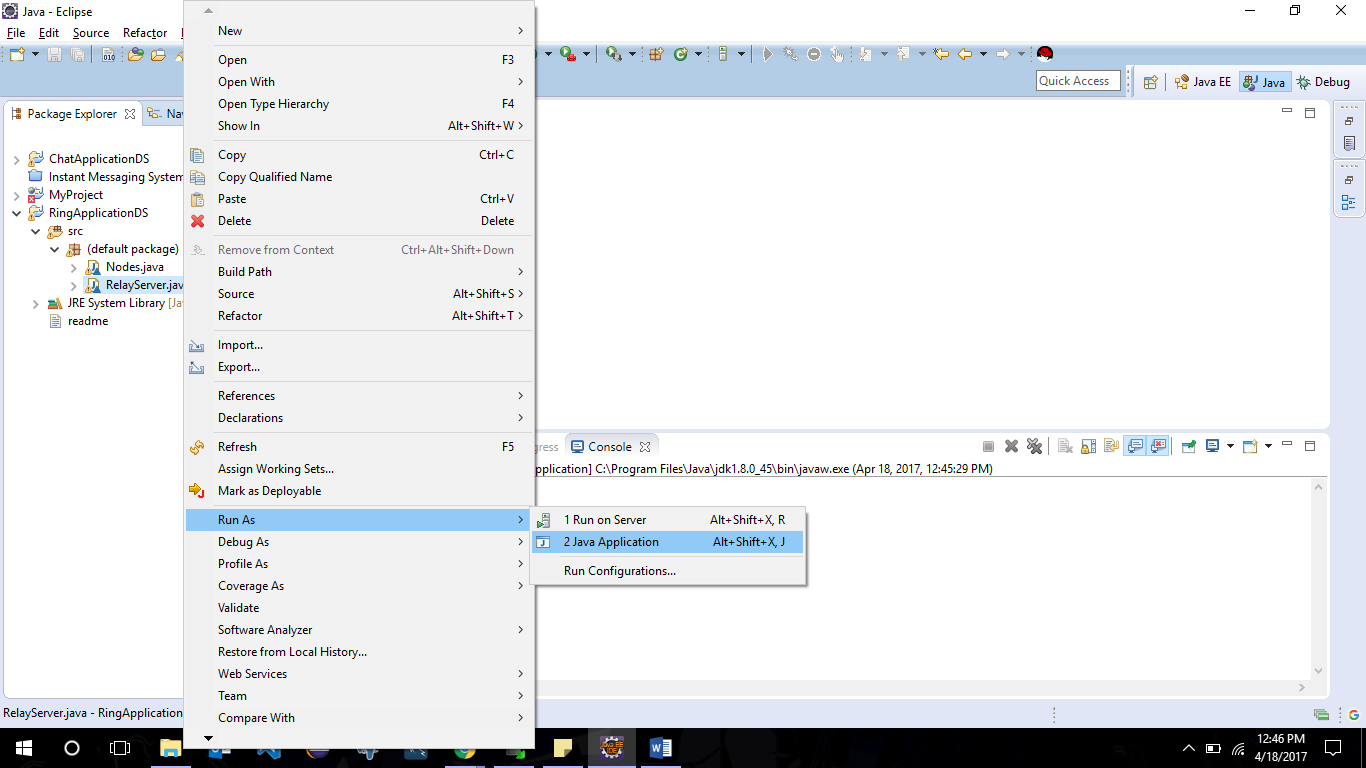
**Extra Functionality (Extra credit functionalities):**

1. All the nodes are represented in the form of GUI.

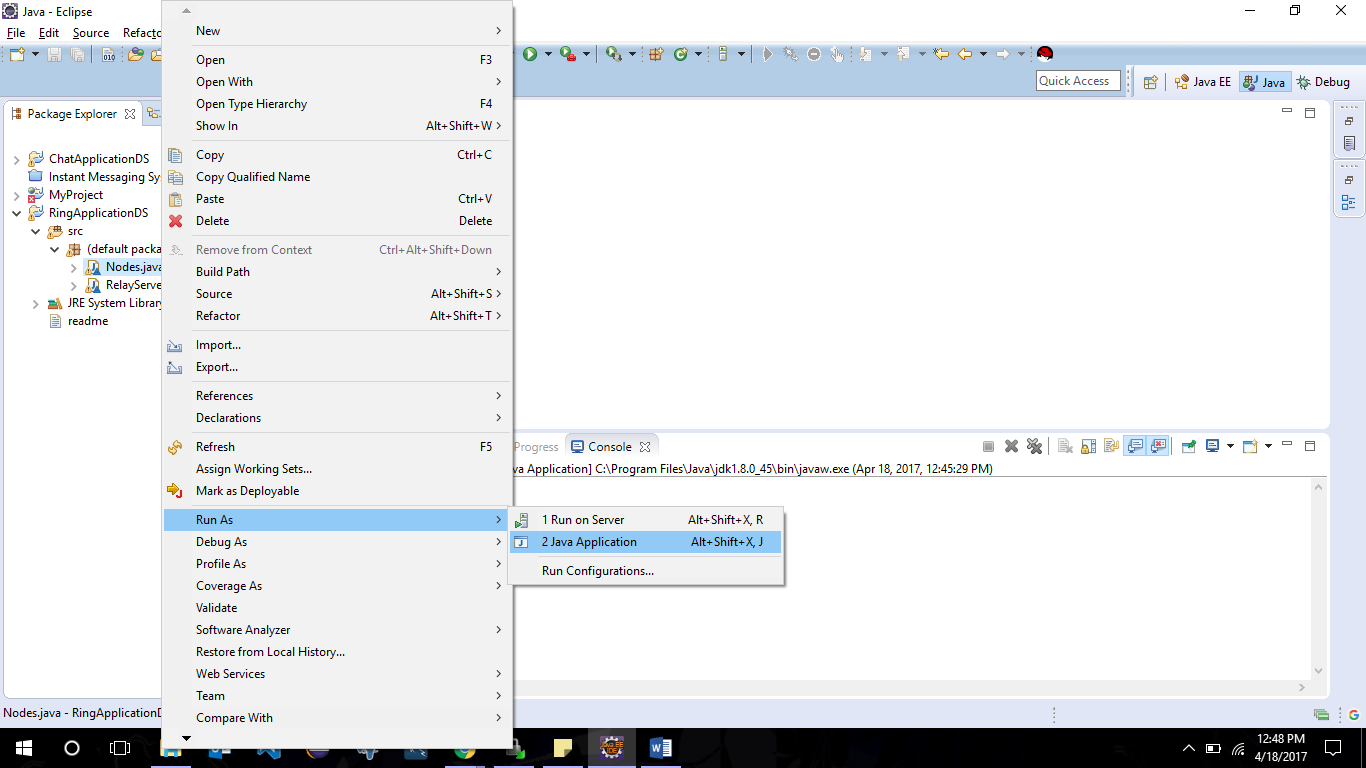
**Screen Shots of the execution process:**

STEP WISE EXECUTION OF THE APPLICATION (SCREEN SHOTS)

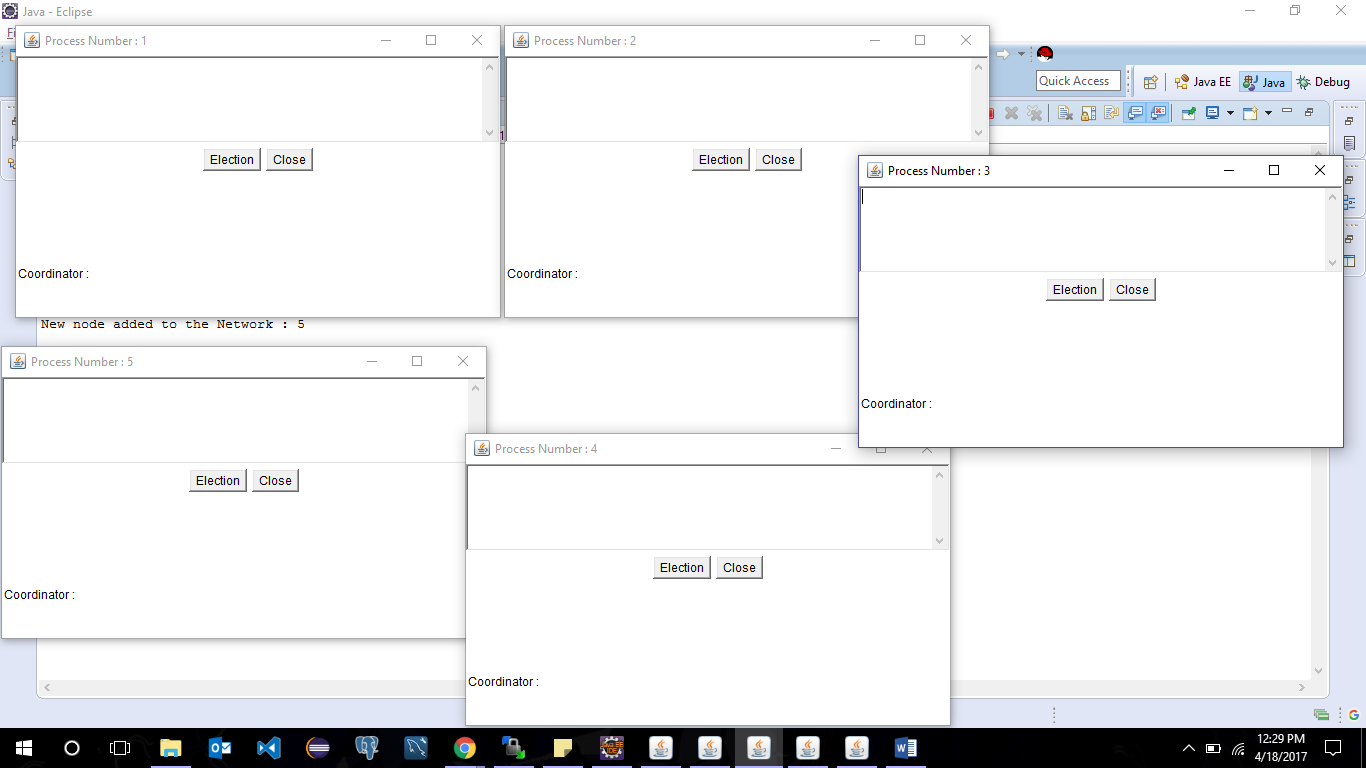
1. Starting the server by running the RelayServer.java as Java Application:



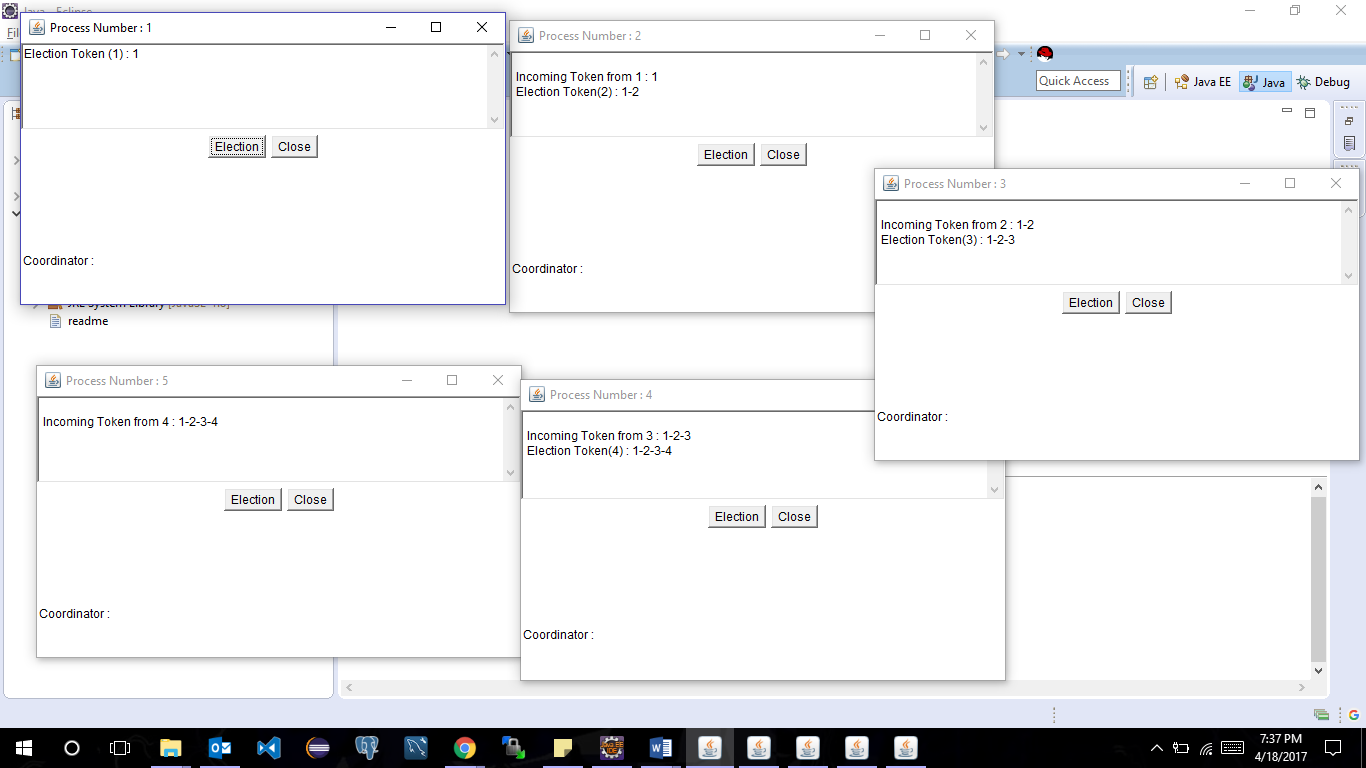
1. Start Nodes by running the nodes.java class as Java Application:



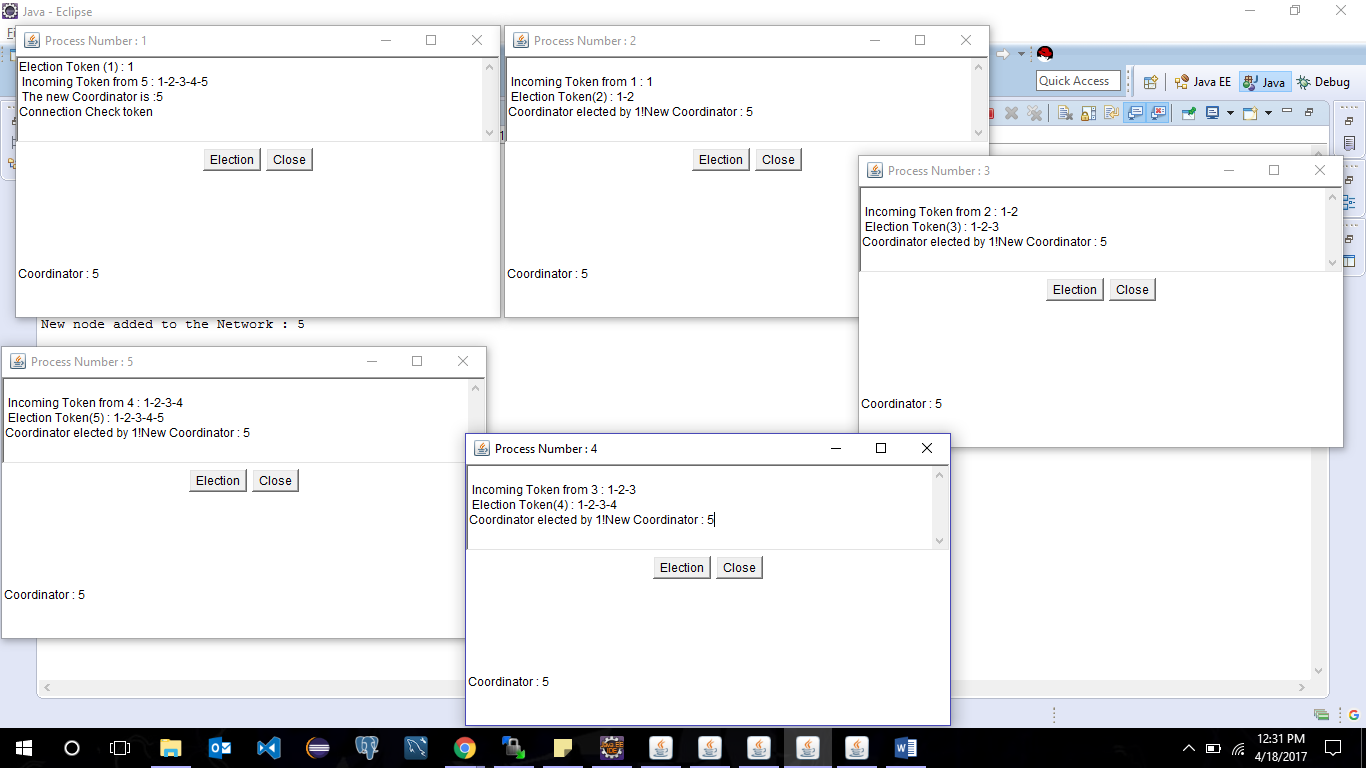
1. Similarly, Open up to five nodes:



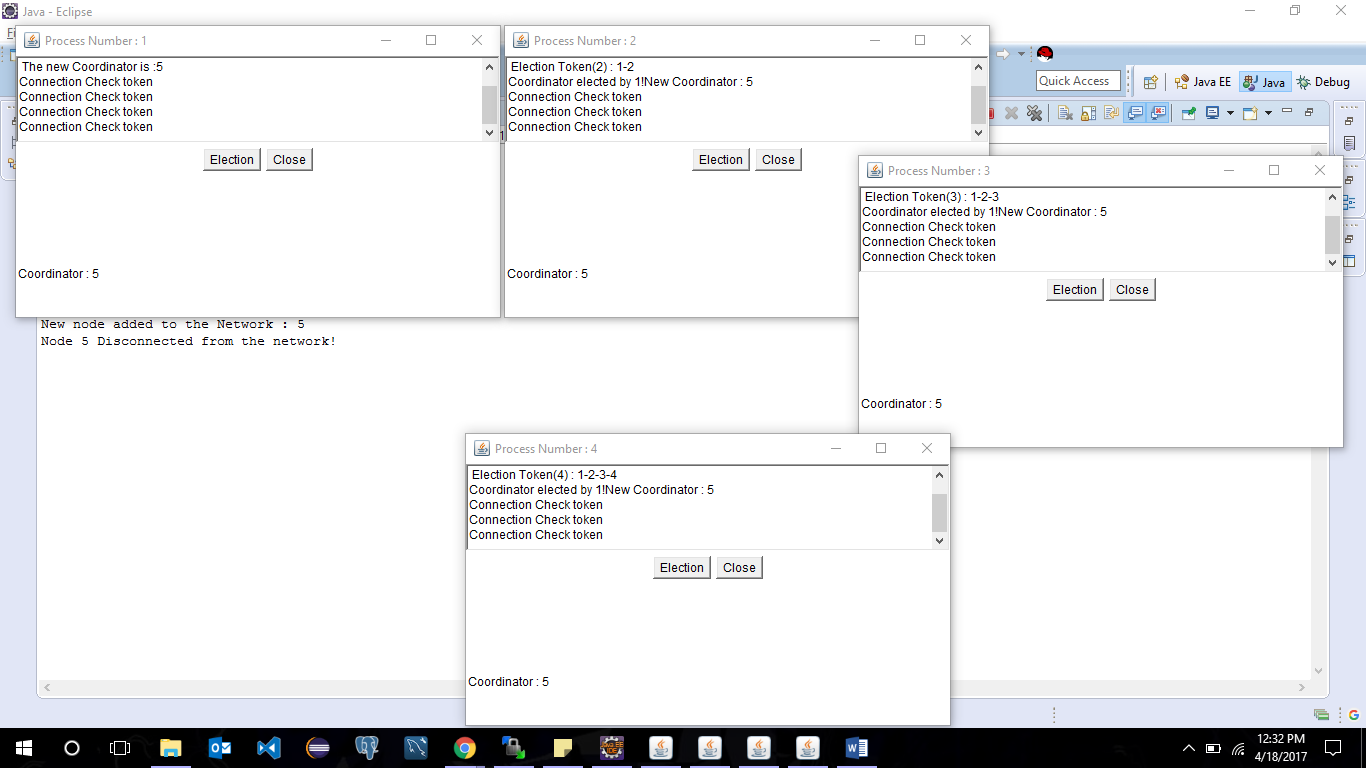
1. Start initial election by clicking the Election button in any one of the nodes(Say 1):



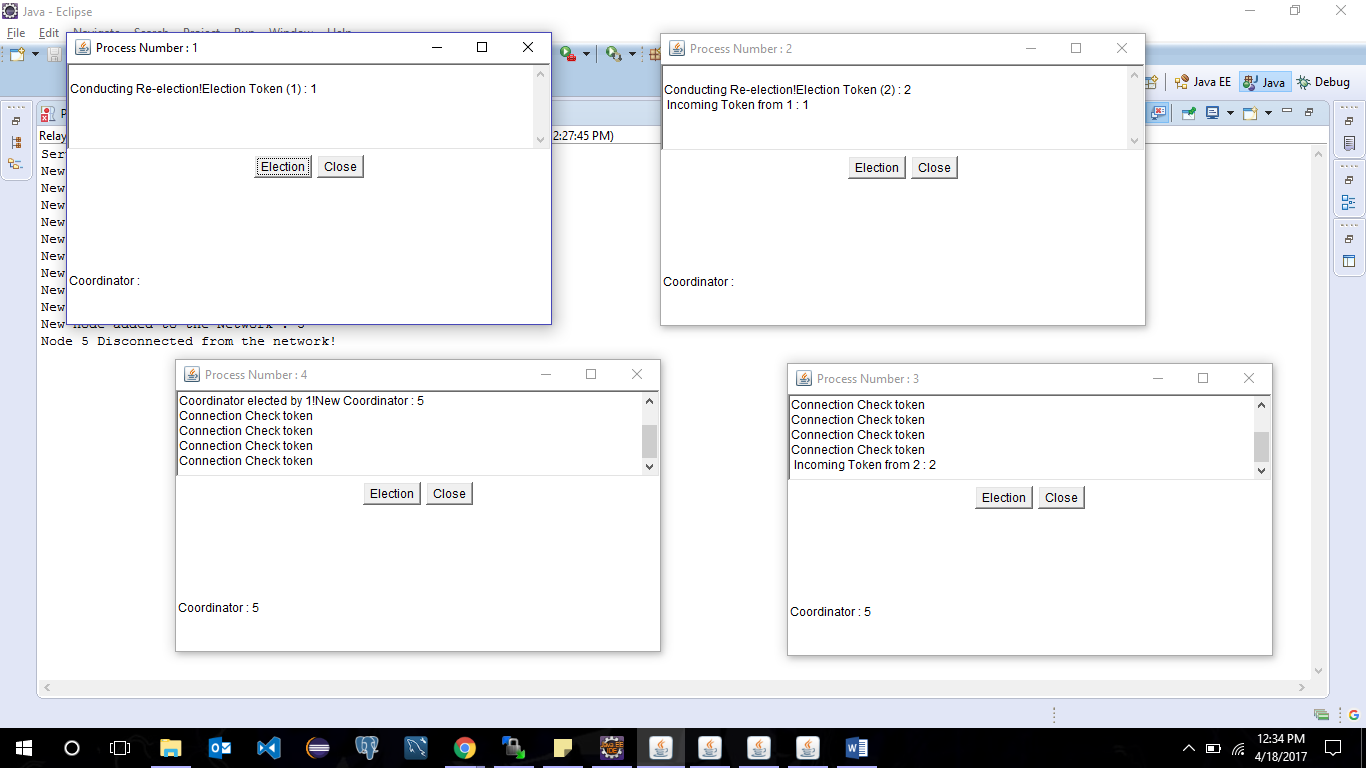
1. We can see that 1 elects the coordinator and informs other nodes about it and starts sending the Connection Check token around. Now our network is setup:



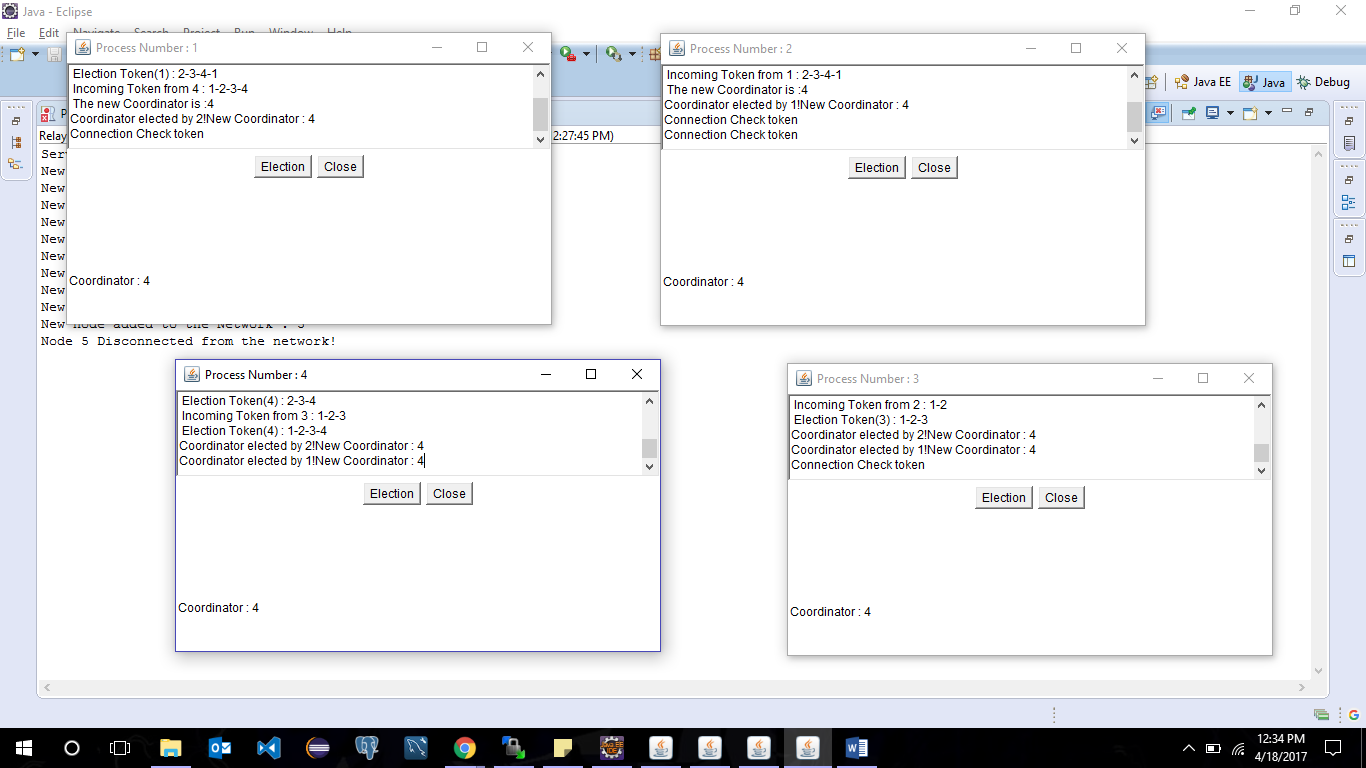
1. Drop coordinator 5 by clicking Close button:



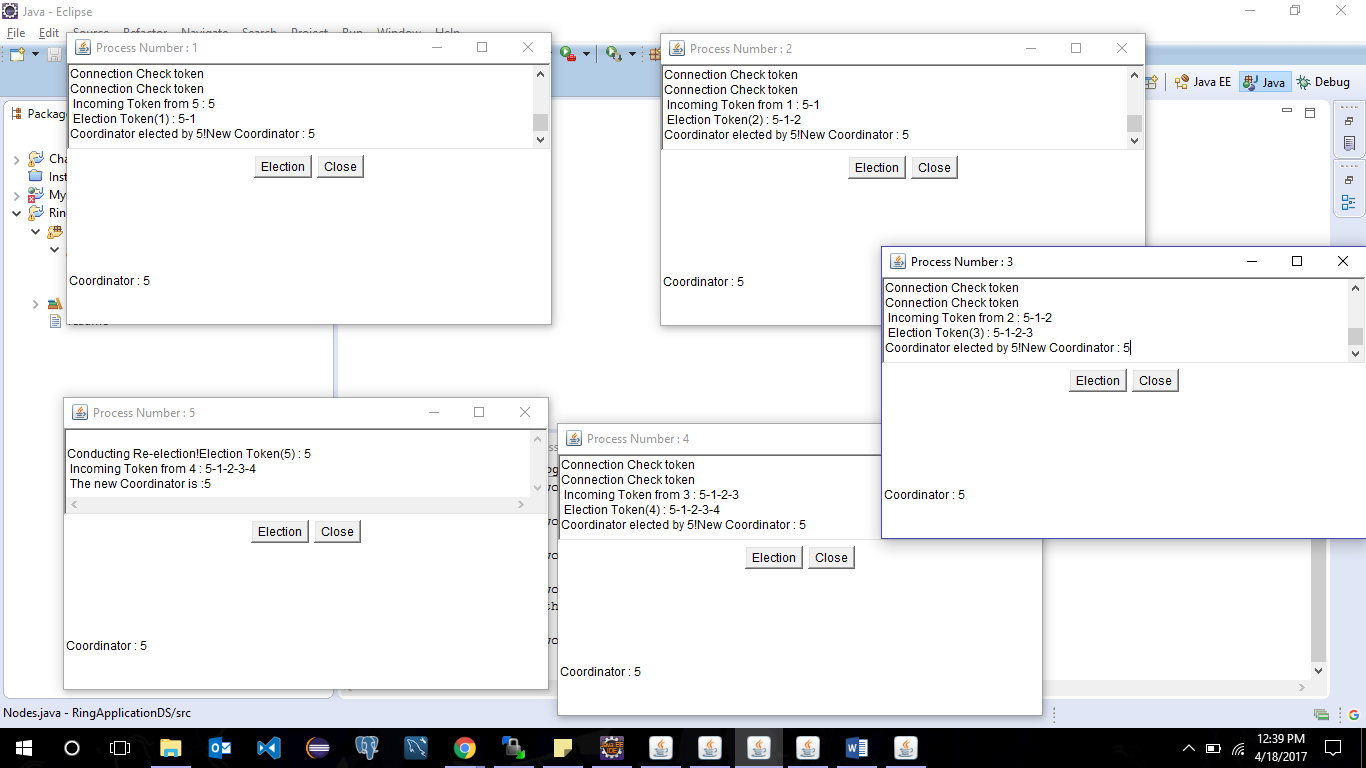
1. After a delay of 50 seconds if the nodes don’t get the Connection check token it starts elections. In our case Node 1 and node 2 both didn’t receive the token for 50 seconds and so its starts the election:



1. And simultaneously both the nodes elections the coordinator and starts Tokens passing:



1. Now we can bring up the coordinator process 5 which we dropped by running the Node.java class as Java application. This node then starts an election and elects the new coordinator:



**Limitations:**

1. Maximum of five nodes can be run at a time. We can run more nodes also but the timer to check whether the token is received or not have to be modified according to the number of nodes. This project is designed with time intervals such that it can handle five nodes appropriately. Only a change in the timer is needed to incorporate more node.

All the requirements are addressed.

Note: Do not stop the server when the client window is up and running. Doing so will make the program go into loop searching for the client socket or thread connection from client side.

**Assumptions:**

1. Node with maximum index value or node number is assumed to be elected as coordinator and all the nodes are arranged in the increment ring fashion in the network based on the node number. Each time we run nodes.java class node will be created with incremental process numbers.
2. Messaging Convention:

* Election token: All election tokens formed at a particular node is represented as “Election Token (node number): value”.
* Incoming election token: All the election token received at a node is such that it mentions the node that sent the token to the current node and along with the token value itself that is nothing but the node number in the list.
* When a coordinator is elected by a node it displays “New coordinator is: node number” and when it circulates this coordinator messages it prints “Coordinator elected by node number! New coordinator: node number”.
* The token that is passed around between the node in ideal scenario is “Connection Check Token”. Ideal meaning when no election is taking place.

**References:**

1. Stack overflow
2. Java Tutorials Point