PRACTICAL-3

AIM:

(1) To change the colour of nodes
(2) To create a network with multiple nodes

Procedure:

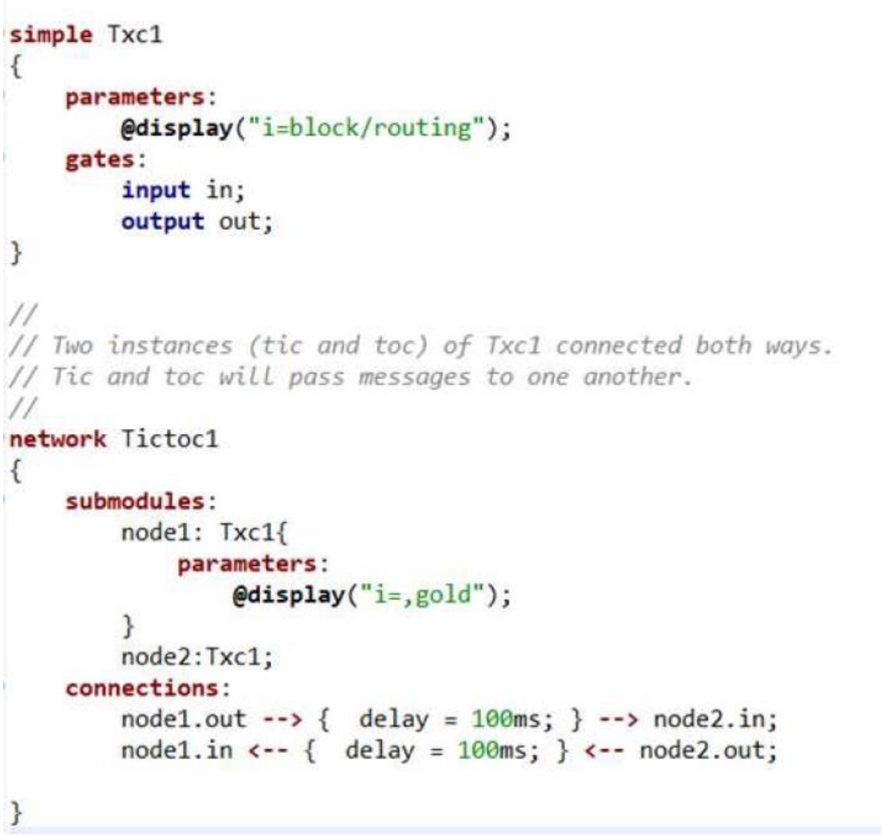
1. Change colour of nodes.

Step 1 : Open the OMNET++ IDE by typing omnetpp in terminal.

Step 2: Create a new project which will be an empty project.

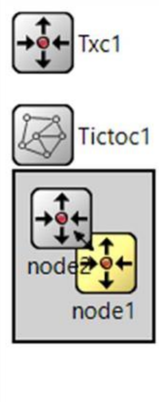
Step 3 : OMNET++ uses NED fies to define components and to assemble them into larger unites like networks. To add the file to the project ,right-click the project directory in the Project Explorer panel on the left, and choose New ->Network Description File (NED) from the menu.

Step 4 : In the source mode of the .ned file, enter the code as in figure.



```
simple Txc1
{
    parameters:
        @display("i=block/routing");
    gates:
        input in;
        output out;
}

//
// Two instances (tic and toc) of Txc1 connected both ways.
// Tic and toc will pass messages to one another.
//
network Tictoc1
{
    submodules:
        node1: Txc1{
            parameters:
                @display("i=,gold");
        }
        node2:Txc1;
    connections:
        node1.out --> {  delay = 100ms; } --> node2.in;
        node1.in <-- {  delay = 100ms; } <-- node2.out;

}
```

Result : Colour of node changed successfully.



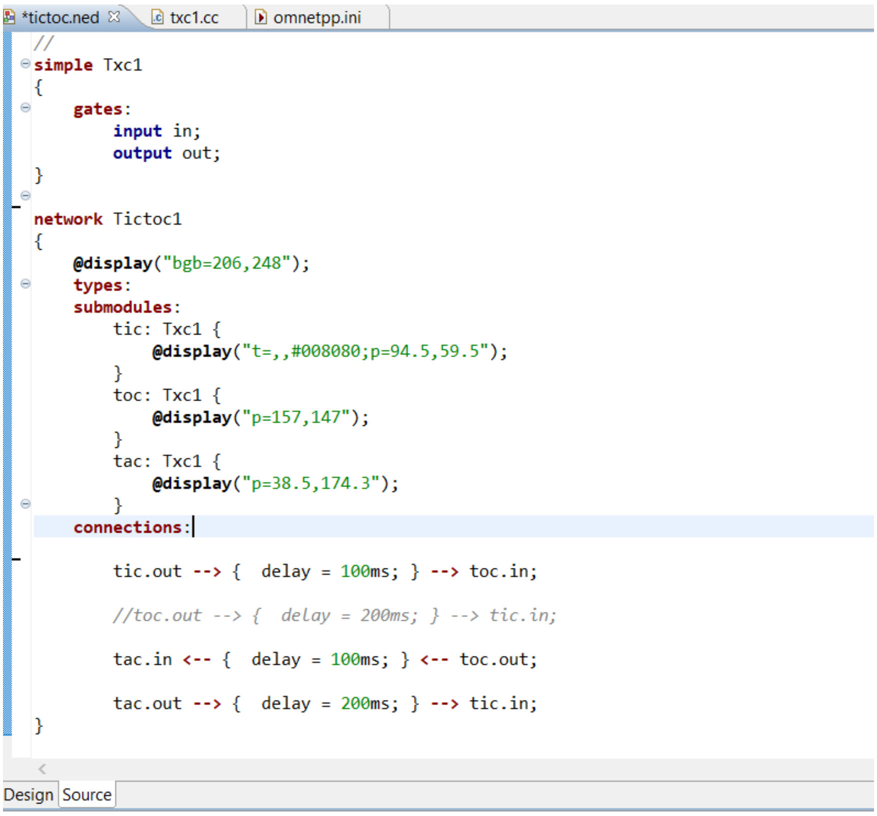
2. To create a network with multiple nodes.

Step 1 : Open the OMNET++ IDE by typing omnetpp in terminal.

Step 2: Create a new project which will be an empty project.

Step 3 : OMNET++ uses NED fies to define components and to assemble them into larger unites like networks. To add the file to the project ,right-click the project directory in the Project Explorer panel on the left, and choose New ->Network Description File (NED) from the menu.

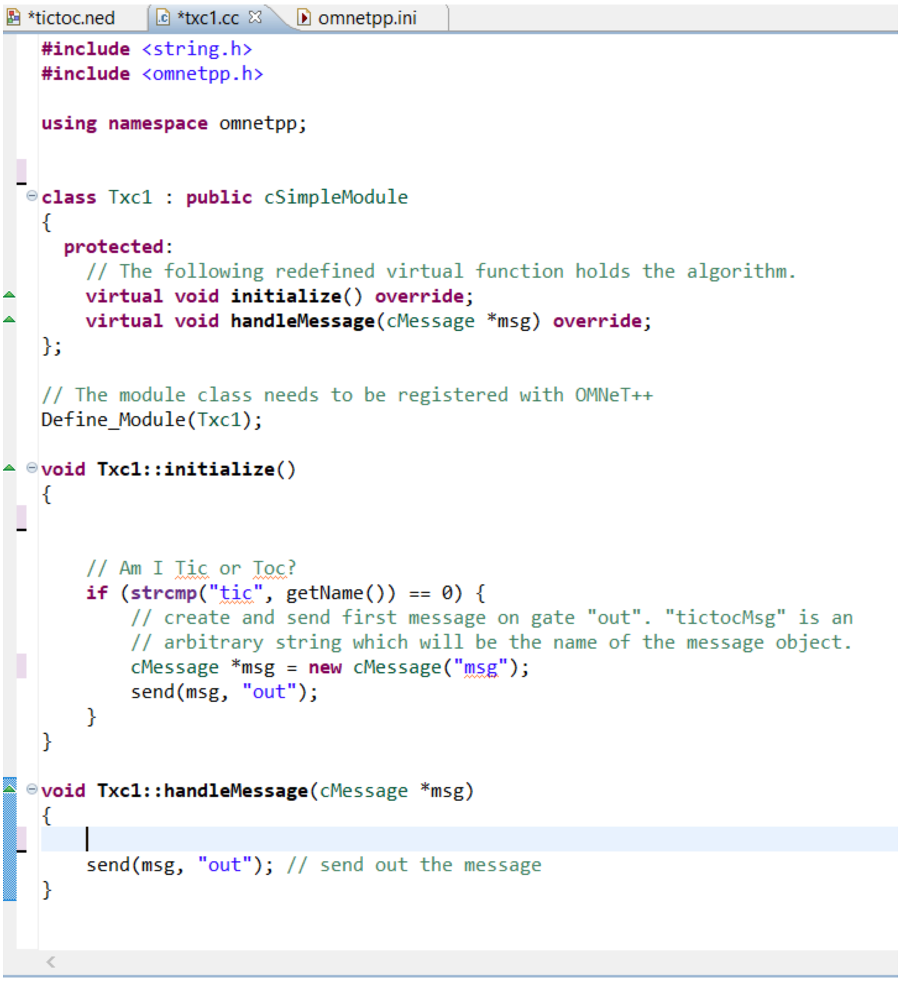
Step 4 : In the source mode of the .ned file, enter the code as in figure.



```
//
simple Txc1
{
    gates:
        input in;
        output out;
}

network Tictoc1
{
    @display("bgb=206,248");
    types:
    submodules:
        tic: Txc1 {
            @display("t=,,#008080;p=94.5,59.5");
        }
        toc: Txc1 {
            @display("p=157,147");
        }
        tac: Txc1 {
            @display("p=38.5,174.3");
        }
    connections:

        tic.out --> {  delay = 100ms; } --> toc.in;

        //toc.out --> {  delay = 200ms; } --> tic.in;

        tac.in <-- {  delay = 100ms; } <-- toc.out;

        tac.out --> {  delay = 200ms; } --> tic.in;
}
```

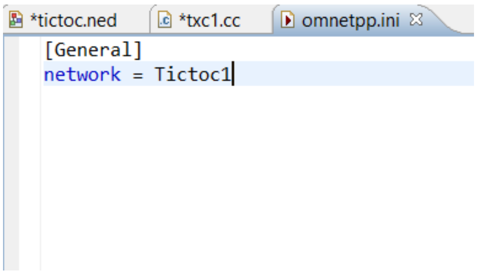
Step 5 : We now need to implement the functionality of the Txc1 module in C++. Create a file by choosing New -> Source File from the project's context menu and give name as per choice.

Step 6 : In the C++ source file(.cc extension) , enter the code as in figure.



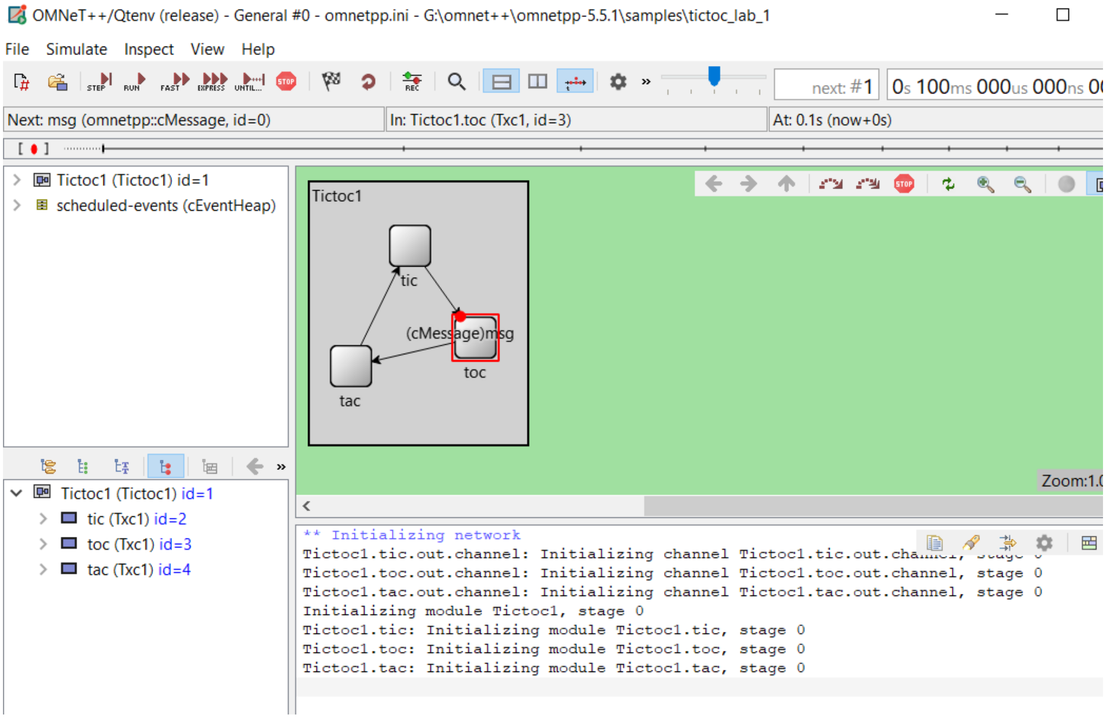
```
#include <string.h>
#include <omnetpp.h>

using namespace omnetpp;


class Txc1 : public cSimpleModule
{
  protected:
    // The following redefined virtual function holds the algorithm.
    virtual void initialize() override;
    virtual void handleMessage(cMessage *msg) override;
};

// The module class needs to be registered with OMNeT++
Define_Module(Txc1);

void Txc1::initialize()
{


    // Am I Tic or Toc?
    if (strcmp("tic", getName()) == 0) {
        // create and send first message on gate "out". "tictocMsg" is an
        // arbitrary string which will be the name of the message object.
        cMessage *msg = new cMessage("msg");
        send(msg, "out");
    }
}

void Txc1::handleMessage(cMessage *msg)
{

    send(msg, "out"); // send out the message
}
```

Step 7: To be able to run the simulation, we need to create an omnetpp.ini file. omnetpp.ini tells the simulation program which network you want to simulate. Create an omnetpp.ini file using the File -> New -> Initialization file (INI) menu item and enter the code as shown.



```
[General]
network = Tictoc1
```

Step 8 : Now the code is complete so click on run button to run the simulator.



OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - G:\omnet++\omnetpp-5.5.1\samples\tictoc_lab_1

Next: msg (omnetpp::cMessage, id=0)

In: Tictoc1.toc (Txc1, id=3)

At: 0.1s (now+0s)

Tictoc1 (Tictoc1) id=1

scheduled-events (cEventHeap)

Tictoc1 (Tictoc1) id=1

- tic (Txc1) id=2
- toc (Txc1) id=3
- tac (Txc1) id=4

```
** Initializing network
Tictoc1.tic.out.channel: Initializing channel Tictoc1.tic.out.channel, stage 0
Tictoc1.toc.out.channel: Initializing channel Tictoc1.toc.out.channel, stage 0
Tictoc1.tac.out.channel: Initializing channel Tictoc1.tac.out.channel, stage 0
Initializing module Tictoc1, stage 0
Tictoc1.tic: Initializing module Tictoc1.tic, stage 0
Tictoc1.toc: Initializing module Tictoc1.toc, stage 0
Tictoc1.tac: Initializing module Tictoc1.tac, stage 0
```

Step 9: Network with multiple node is created is implemented successfully.