BTN415 Lab 5 – OO Paradigm Review

In this lab, you will create the declarations of classes, as well as basic constructor and destructor functions, to serve as blueprint to TCP/IP server and client objects.

# LEARNING OUTCOMES

Upon successful completion of this lab, you will have demonstrated the ability to:

* Create inherited classes
* Create constructors and destructors

For this lab, you should create two files, one called *socket.h* and other called *socket.cpp*. Then, you can test your results by running them with the main source code file which is provided in: <https://github.com/marceljar/BTN415_Labs/blob/main/lab5/main.cpp>.

## PART A – [2.0 marks]

## Create the declaration of a Node class with the following properties and methods. Note that you do not have to really implement the methods. They should all simply print their own names, as well as the list of inputs that they have received as arguments. For example, a method called start\_dlls() should simply print “start\_dlls was called”, and send\_message(“hello”) should print “send message was called with: hello”.

### Properties (all protected):

* A static Boolean variable called *dlls\_started*
* A static int variable called *num\_nodes*
* A string variable called *ip*
* A string variable called *role*
* A string variable called *protocol*
* An int called *port*
* A SOCKET variable called *active\_socket*

### Methods (all public):

* A base constructor
* A constructor that takes two arguments: a string (for the *ip*) and an int (for the *port*)
* A destructor
* A method called *start\_dlls* that does not alter any properties of Node, takes no arguments, and returns nothing
* A method called *create\_socket* that takes no arguments and returns a Boolean
* A method called *display\_info* that takes no arguments, does not alter any properties of Node, and returns nothing

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## PART B – [1.5 marks]

## Create the declaration of a Server\_TCP class, derived from Node, with the following properties and methods:

### Properties (all private):

* A SOCKET variable called client*\_socket*

### Methods (all public):

* A base constructor
* A constructor that takes two arguments: a string (for the *ip*) and an int (for the *port*)
* A destructor
* A method called bind\_socket that takes no arguments and returns a Boolean
* A method called listen\_for\_connections that takes no arguments and returns a Boolean
* A method called accept\_connections that takes no and returns a Boolean
* A method called send\_message that takes a string argument (for a message) and returns an integer
* A method called receive\_message that takes a reference for a string argument (for a message) and returns an integer
* A method called close\_connection that takes no arguments and returns nothing

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## PART C – [1.5 marks]

## Create the declaration of a Client\_TCP class, derived from Node, with the following methods:

### Methods (all public):

* A base constructor
* A method called connect\_socket that takes two arguments: a string (for the *ip*) and an int (for the *port*) and returns a Boolean
* A method called send\_message that takes a string argument (for a message) and returns an integer
* A method called receive\_message that takes a reference for a string argument (for a message) and returns an integer.

Text

Description automatically generated

Text

Description automatically generated

**Part A + Part B + Part C – Output**

Graphical user interface

Description automatically generated

### SUBMISSION INSTRUCTIONS

*You only need to submit two source codes, a declarations socket.h file, and an implementation socket.cpp file.*