BTN415 Lab 7

Transmitting Structures with Dynamically Allocated Items

In this lab, you will learn how to transmit structures containing both static, as well as dynamically allocated fields of data over sockets.

LEARNING OUTCOMES

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

1. Transmit and receive structures containing static fields and dynamically allocated fields

SPECIFICATIONS

In this lab, we will create methods to enhance our **oop_winsock** library, so that it can send data containing dynamically allocated data. Currently, as our starting point (available on Github) this library can only be used to send **char arrays**. The methods that will need to be created are described in what follows.

oop_winsock_client::send_frame, oop_winsock_server::receive_frame

```
struct frame {
      int length;     //represents the number of elements in the body
      int *body;
      int tail;
};
```

The **send_frame** method should take as an argument a **struct** of type **frame** (defined above). It should copy all of its data, including the dynamically allocated data, into a local **char array** buffer, and send this buffer over a tcp socket. This method should not return any values, but should use the Print method to display the content of the frame.

The **receive_frame** method should take as an argument a **struct** of type **frame** (defined above). This argument should be passed by reference. The received data should be first stored into a **char** array buffer. Following this method should copy the received data into the **frame struct** passed as an argument. This method should not return any values, but should use the Print method to display the content of the frame.

TAKE HOME

Update your solution to allow the two way communications. Your server should create a new frame that contains your name stored in the Body and transmit it back to the client as a response.

NOTE: The send_frame and receive_frame methods will not display your name correctly. Instead it will print out large integer numbers. You should verify from the output.txt files that your data being transmitted matches that which was received.

SUBMISSION REQUIREMENTS

Once you have completed your lab create and upload the following files:

- Create a single ZIP file that contains all your source code files (*.h and *.cpp)
- The output.txt files generated by the lab
- Any additional information you feel necessary for me to mark your lab