BTN415 Lab 6

Transmitting Arrays, Structs, and Structs with Bitfields

In this lab, you will learn how to transmit different types of data over sockets.

LEARNING OUTCOMES

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

1. Transmit and receive arrays of floats with a defined size over sockets
2. Transmit and receive regular structs over sockets
3. Transmit and receive structs containing bitfields

SPECIFICATIONS

In this lab, we will create methods to enhance our **oop_winsock** library, so that it can send data with different types of formats. 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_float_array, oop_winsock_server::receive_float_array

The **send_float_array** method should take as an argument an array of type **float**, copy it into a local **char** array buffer, and send it over a tcp socket. The array of type **float** should have a constant size of **4** (four) elements. This method should not return any values.

The **receive_float_array** method should take as an argument an array of type **float**. It expects an array of type **float** with size **4** (four) elements to be received through a socket connection. The received data should be first stored into a **char** array buffer. Following this method should copy the received data into the float array provided as an argument[1]. This method should not return any values.

oop_winsock_client::send_packet, oop_winsock_server::receive_packet

```
struct packet {
      int student_number;
      char[32] student_name;
      float student_gpa;
};
```

The **send_packet** method should take as an argument a **packet** struct (defined above). The input packet should be used directly as an argument to the **send() winsock** method. This method should not return any values. *Hint casting*.

[1] Remember that arrays are, by default, passed by reference in C++. Hence, if you modify an array passed as an argument to a function, the modification will continue to affect the array after the function finishes execution.

The **receive_packet** method should take as an argument a **packet** struct (defined above). This argument should be passed by reference. The method expects a **packet** struct to be received through a socket connection. The **recv() winsock** should take as an argument the packet passed as an argument In other words, do not use an intermediate **char array buffer**. This method should not return any values. *Hint casting*.

oop_winsock_client::send_bitpacket, oop_winsock_server::receive_bitpacket

```
struct bitpacket {
      int student_number;
      char[32] student_name;
      unsigned char current : 1;
      unsigned char doing_coop : 1;
      unsigned char academic_violations: 1;
      unsigned char : 5; //padding
      float student_gpa;
};
```

The **send_bitpacket** method should take as an argument a **bitpacket** struct (defined above). The input bitpacket should be used directly as an argument to the **send() winsock** method. This method should not return any values. *Hint casting*. *Hint2: This struct has a total size of 44 bytes.*

The **receive_bitpacket** method should take as an argument a **bitpacket** struct (defined above). This argument should be passed by reference. This method should first use a **char array buffer** to collect a **bitpacket** transmitted over a **winsock** connection. Then, this method should use pointer arithmetics, together with bitwise operations, to paste the contents of the receiving **char array buffer** into the **bitpacket** passed as an argument. If you simply use **memcpy** for your **receive_bitpacket** method, instead of pointer arithmetics, you will lose marks.

Take Home

Update the following functions to display the contents of the data packets using the built in **Print()** method.

- Winsock_server
    - void send_float_array(float *);
    - void receive_float_array(float *);
    - void receive_packet(packet &); *HINT: Thinking timing*
    - void receive_bitpacket(bitpacket &);
- winsock_client
    - void send_float_array(float *);
    - void receive_float_array(float *);
    - void send_packet(packet);
    - void send_bitpacket(bitpacket);

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