BTN415 Test 2 - Programming

Student Name: **Soham Thaker** Student ID: **011-748-159**

Design complete handshaking client-server over TCP protocol using port 12700 by sending a serialized packet with CRC and Checksum. The server will validate any packet from the client, if this packet is valid the server will reply by ack packet.

## PART A – [1.5 marks]

## Create a function, called create\_packet(), that will ask the user for all required information to create a new packet, which has the structure shown below:

struct data\_packet {

int seq;

unsigned char ack : 1;

unsigned char syn : 1;

unsigned char flag : 1;

unsigned char rst : 1;

unsigned char : //padding

unsigned int length;

char\* data;

unsigned char crc;

};

Note that the fields **length** and **crc** should be inferred from the data provided by the user. First, by checking the **strlen()** of the data entered by the user, to determine its length. And then, by calculating the **crc** with the **calculate\_data\_packet\_crc()** function that is provided in the repository.

**---------- Screenshot of Packet Code --------**

Text

Description automatically generated

## PART B – [2.5 marks]

### serialized\_packet data\_packet\_serializer(new\_packet)

This function (which should not belong to any class, but rather be a regular function), should take as an argument a **data\_packet**. Then, as the name suggests, it should save all contents of the **data\_packet** (including bit-fields and dynamically allocated ones) into a *char array*. Finally, the function should return a **serialized\_packet** struct containing the char array as well as the length of this char array (i.e., the total size of the packet in bytes).

**---------- Screenshot of Serialize Packet Code --------**

## PART C – [2.5 marks]

Text

Description automatically generated

### data\_packet new\_packet\_deserializer(char \*)

This function (which should not belong to any class, but rather be a regular function), should take as an argument a *char array*. As the name suggests, this function should save all contents of the *char array* in the proper fields inside a **data\_packet** struct which, in turn, it then returns as an output.

**---------- Screenshot of Deserialize Packet Code --------**

Text

Description automatically generated

**--- Screenshot of OUTPUT Serialize Packet --- Screenshot of OUTPUT Dserialize Packet----**

## Text Description automatically generated

## PART D – [1.5 marks]

### calculate\_crc(data\_packet)

You need to build a function to calculate and validate CRC based on the below polyamine:

Diagram

Description automatically generated

**---------- Screenshot of Calculation of CRC Code --------**

*Text

Description automatically generated*

**---------- Screenshot of Calculation of CRC OUTPUT --------**

**CRC output for client code:**

## Text Description automatically generated

## PART E – [2.0 marks]

### int send\_data\_packet (data\_packet) (for both the Server\_TCP as well as Client\_TCP classes)

This method, as its name suggest, should send a **data\_packet** over a socket and return the number of transmitted bytes. *Hint: Use the data\_packet\_serializer() function to accomplish your goals.*

**---------- Screenshot of Send Data Packet Code --------**

*Text

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated*

### int receive\_data\_packet (data\_packet&) (for both the Server\_TCP as well as Client\_TCP classes)

This method, as its name suggest, should receive data sent over a socket, and save it in the proper fields of a **data\_packet** struct. It should also return the number of received bytes. *Hint: Use the data\_packet\_deserializer() function to accomplish your goals.*

**---------- Screenshot of Receive Data Packet Code --------**

# Text Description automatically generated

Text

Description automatically generated

**--- Screenshot of OUTPUT Sending Packet --- Screenshot of OUTPUT Receive Packet----**

**You need to show also the CRC and Validation of CRC**

# Text Description automatically generated

# SUBMISSION INSTRUCTIONS

Submit your client.cpp, server.cpp, socket.cpp and socket.h files and This file with all screenshots.