Exercise 3

ClientTranslationApplication.java

```
package demotcpclientText;

import java.io.*;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Scanner;

/**
* This is client side application
*
* @author Zaki ARMINDO
*
*/
public class ClientTranslationApplication {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		try {

			//Connect to the server at localhost, port 4228
			Socket socket = new Socket(InetAddress.getLocalHost(),4228);

			//Create input stream
			BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

			//Read from the network and display the date
			String text = bufferedReader.readLine();
```

```
                        System.out.println(text);

                        //Close everything
                        bufferedReader.close();

                        socket.close();


                } catch(IOException e) {

                        e.printStackTrace();
                }

        }

}
```

ServerTranslationApplication.java

```
package demotcpserverText;

import java.io.IOException;
import java.io.PrintStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
```

Zaki ARMINDO(B031910455)

```
/**
 * This  server class will count the number of words in the
 *  text and returns it to the client..
 *
 * @author Zaki ARMINDO
 *
 */
public class ServerTranslationApplication {

	public static void main(String[] args, String words) throws IOException{

		ServerSocket serverSocket = null;

		try {
			//Bind Server socket to a port
			int portNo = 4228;
			serverSocket = new ServerSocket(portNo);

			String text1 = "Good Afternoon";
			System.out.println("Waiting for request");
			int count = 0;
			while(true) {

				//Accept client request for connection
				Socket clientSocket = serverSocket.accept();

				//Create stream to write data on the network
				DataOutputStream outputStream = new
DataOutputStream(clientSocket.getOutputStream());

				//Send current data back to the client
```

```
                    outputStream.writeUTF(text1);



                    //call method
                    count=countWord(words);

                    outputStream.writeInt(count);
                    outputStream.flush();

                    //Close the socket
                    clientSocket.close();
                }


            }catch (IOException ioe) {

                if(serverSocket != null)
                    serverSocket.close();

                    ioe.printStackTrace();
            }



        }

        public static int countWord(String words)
        {
         if(words.isBlank())
         {
         return 0;
         }
```

```
        int count = 0;


        for(int index = 0;index<words.length() -1 ;index++)
        {
        if(words.charAt(index) == ' ' && words.charAt(index +1) != ' ')
        {
        count += 1;
        }
        }
        return count + 1;
       }


}
```

Exercise 5

LanguageTranslationSever.java

```
package lab4_client;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Scanner;

public class LanguageTranslationClient {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		try {
			//Connect to the server at localhost for port 4228
			Socket socket = new Socket(InetAddress.getLocalHost(),4228);

			//Create input object
			Scanner sc = new Scanner(System.in);

			//create output stream,
			DataOutputStream outputStream = new
			DataOutputStream(socket.getOutputStream());

			//Display
			System.out.println("-----------------------------------------");
			System.out.println("\t Client Side");
			System.out.println("-----------------------------------------");
```

```
                System.out.println(" ");
                //Input From User
                System.out.print("Enter Words In English: ");
                String text =(sc.nextLine());
                //send data to server
                outputStream.writeUTF(text);
                outputStream.flush();

                //create input stream
                DataInputStream dataIn = new DataInputStream(socket.getInputStream());

                //assign object to read data from server
                String translatedText = dataIn.readUTF();
                System.out.println("\nTranslated text: " );
                System.out.println("\nMalay      Arabic     Korean");

                //print translated text sent from server
                System.out.println(translatedText);

                //Close everything
                outputStream.close();
                sc.close();
                socket.close();
                dataIn.close();
                } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
                }


        }
```

```
}
```

LanguageTranslationClient.java

```
package lab4_client;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Scanner;

public class LanguageTranslationClient {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		try {
			//Connect to the server at localhost for port 4228
			Socket socket = new Socket(InetAddress.getLocalHost(),4228);

			//Create input object
			Scanner sc = new Scanner(System.in);

			//create output stream,
			DataOutputStream outputStream = new
			DataOutputStream(socket.getOutputStream());

			//Display
			System.out.println("-----------------------------------------");
```

```
System.out.println("\t Client Side");
System.out.println("-----------------------------------------");
System.out.println(" ");
//Input From User
System.out.print("Enter Words In English: ");
String text =(sc.nextLine());
//send data to server
outputStream.writeUTF(text);
outputStream.flush();

//create input stream
DataInputStream dataIn = new DataInputStream(socket.getInputStream());

//assign object to read data from server
String translatedText = dataIn.readUTF();
System.out.println("\nTranslated text: " );
System.out.println("\nMalay      Arabic     Korean");

//print translated text sent from server
System.out.println(translatedText);

//Close everything
outputStream.close();
sc.close();
socket.close();
dataIn.close();
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
```

```
        }

}
```

Translator.java

```
package lab4_server;

public class Translator {

        private String text;

        public Translator(String text) {

                this.text=text;
                //this.language=language;
        }

        public Translator() {
        }

        public void setText(String text) {
                this.text=text;
        }

        public String getText() {
                return text;
        }

        public String getTranslatedText() {
        String translatedText = "";
```

```
        if(text.contentEquals("Good Morning")||text.contentEquals("Good
morning")||text.contentEquals("good morning")) {

         translatedText="Selamat Pagi الخير 좋은 아침";

        }


        else if (text.contentEquals("Good Night")||text.contentEquals("Good
night")||text.contentEquals("good night")) {translatedText="Selamat malam مساء الخير 안녕히
주무세요";

        }


        else if (text.contentEquals("How are you?")||text.contentEquals("how are you?"))
{translatedText="Apa khabar? كيف حالكم؟ 어떻게지내세요?";

        }


        else if (text.contentEquals("Thank you")||text.contentEquals("thank you"))
{translatedText="Terima kasih شكرا جزيال 감사합니다";

        }
        else if (text.contentEquals("Goodbye")||text.contentEquals("goodbye"))
{translatedText="Selamat Tinggal مع السالمة 안녕";

        }


        else if (text.contentEquals("What's up?")||text.contentEquals("what's up?"))
{translatedText="Ada apa? ما أخبارك؟ 뭐야?";

        }
        return translatedText;

         }



}
```

Zaki ARMINDO(B031910455)