



**School of Computing**

**Bachelor of Science in Computing.**

**Programme Code: DT211/3**

**2012 – 2016**

Network Programming

**Lecturer: Mark Deegan**

<b>Student's Name</b>	James Wilson
<b>Class Group</b>	DT211/3
<b>Assignment Number</b>	1
<b>Assignment Title</b>	Document
<b>Date Issued</b>	17/1/2015
<b>Date Due For Return</b>	13/2/2015
<b>Date Returned</b>	13/2/2015

## How did I run the program?

To develop this program TxtClock I complied it in command line, using javac command (which compiles the program into byte code ) to compile the java file for example when I am in the appropriate directory I run the command `javac -d .\bin\ .\src\*.java`. Then when I am running the java file in the src directory, the command I use is `java -cp ..\bin ie.dit.student.wilson.james.TxtClock` (arguments passed through if any).

## What did I change in order for the program to be more efficient?

In my program, I changed the code quite a bit. Firstly it took me while to figure out what way I would store all the words in an array without it looking too bulky and inefficient. In my first design I had an array of words from 0-59, also I had a tonne of if statements. If input == 1 etc, but it took up too many lines, and designed and implemented algorithms that would execute the same purpose but in less lines of code. I stored them into two string arrays, the first one being all the tens, ten twenty etc, and the other be the unit's so to speak one to nineteen. Using the correct operators I was able to develop a method where if the number is below 20 it would execute the else statement and that number passed through the function (hour, minute, or second) and that number modulus 10 (remainder of 10) in the array it would select.

## NTP Server

In my program I was also having trouble connecting to the NTP server. I researched a lot about NTP servers, I learned and understood how and what port they connected through via the internet. Ports 13 and 123 are the two main protocols that NTP clients communicate through. The one used for my the day time port 13.

Once I was to connect to a time server, (using a socket connection) over the input Stream which java was able to put together the bytes into packets sending them and receiving them.

I was able to ping to the server, and it would acknowledge that it was online, and when it was offline, but I done it in a try and catch instead, as it was functioning perfect, when it couldn't reach the server it read for the local system.