Trenton Samuel

**ECE TCP/IP Lab**

**Email Java Project**

**Objective**

My objective of this project is to run a Java program that sends an email from a hard-coded, specified email address to another hard-coded, specified email address. I also want to run a Java program that sends an email from an application, which allows the SMTP (Simple Mail Transfer Protocol) server, email addresses, the subject, and the message to be changed without having to rerun the program. Lastly, I want to run a Java program that sends an email from an applet, which is similar to sending an email from an application, except with the applet, the email address that receives the email and the SMTP server can only be changed by editing the program’s code.

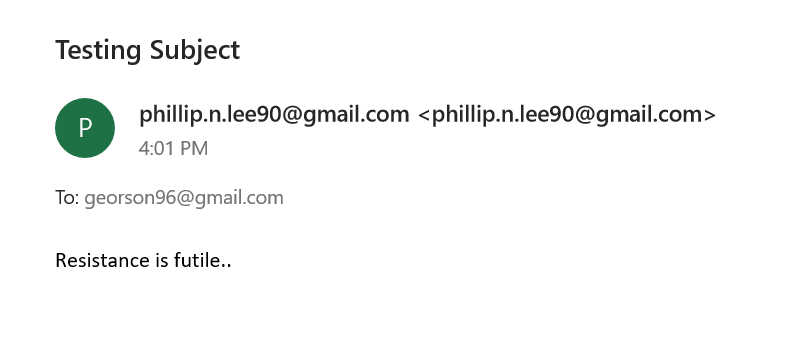
**Materials**

* Two computers with access to WAN.

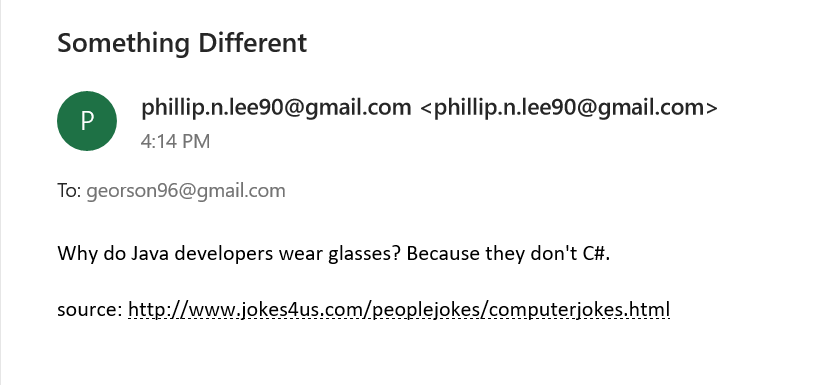
**Part I**

In Part I of this experiment, I set out to run a program called assimilator.java that will send out an email (the sending and receiving email addresses, the subject, the message, and the SMTP server are all hardcoded).

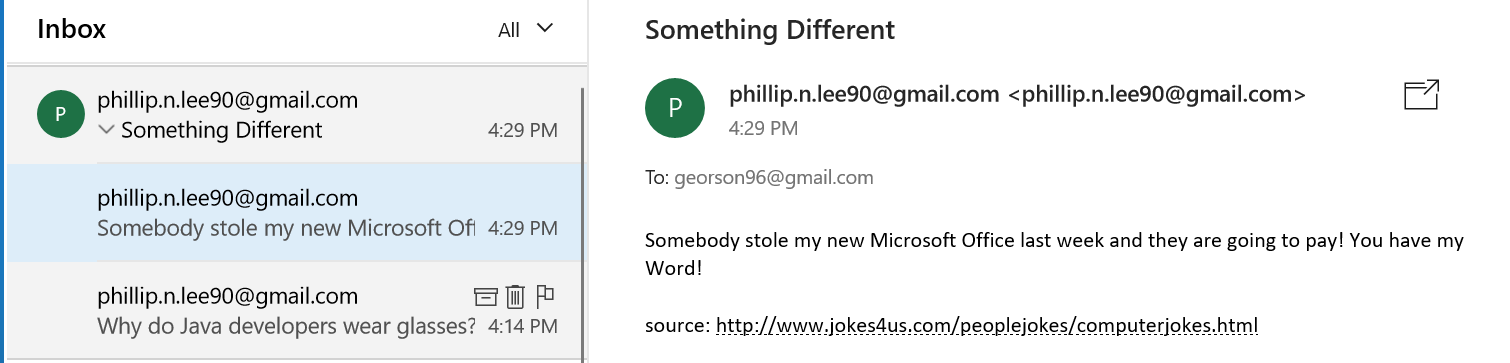
***Steps 1-4:*** First, I created a Java project folder that would contain all programs that I would end up using for all parts of this experiment. I then created a program utilizing the Example 19.1 (assimilator.java) code from the Java Network Programming textbook provided by my professor. I next downloaded the Java archive files of javax.mail.jar and activation.jar and added them to my project (This involved selecting the Project tab, clicking Properties, selecting Java Build Path on the left side bar of the pop-up box, and clicking the Add JARS button). I lastly modified the assimilator.java code so that my program would be able to send out an email (I altered the code so that my program would be able to function with smtp.gmail.com as my SMTP server).

Below is the result of running my program so that an email with the message of “Resistance is futile..” is sent from a classmate’s (Gmail) email address to my (Gmail) email address.  
  


***Steps 5-7:*** I then ran the program a few more times, except now changing (within the code itself) a few aspects of the email that will be sent out, such as the subject line and message as shown below.



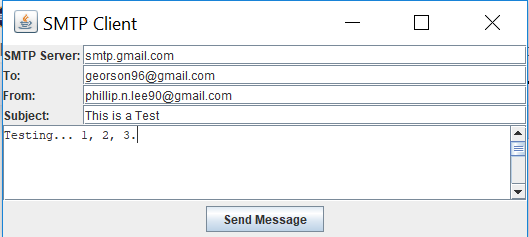
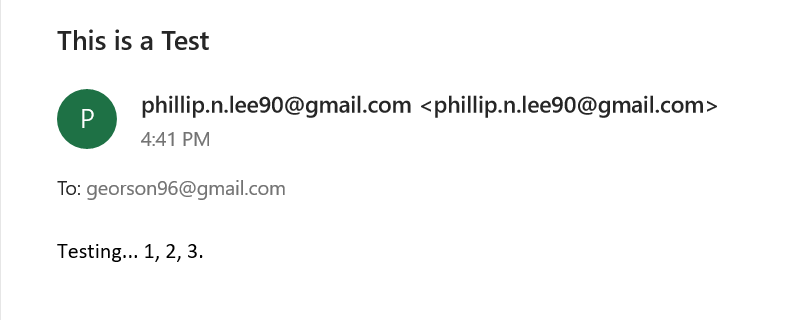
Below shows how I ran the program sending an email with the same subject as the previous email (from the previous screenshot), but with a different message.

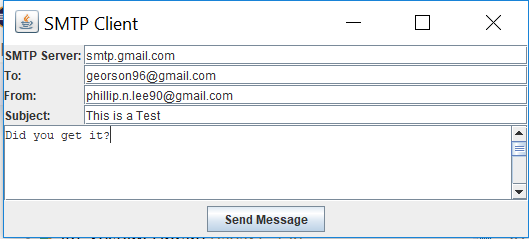
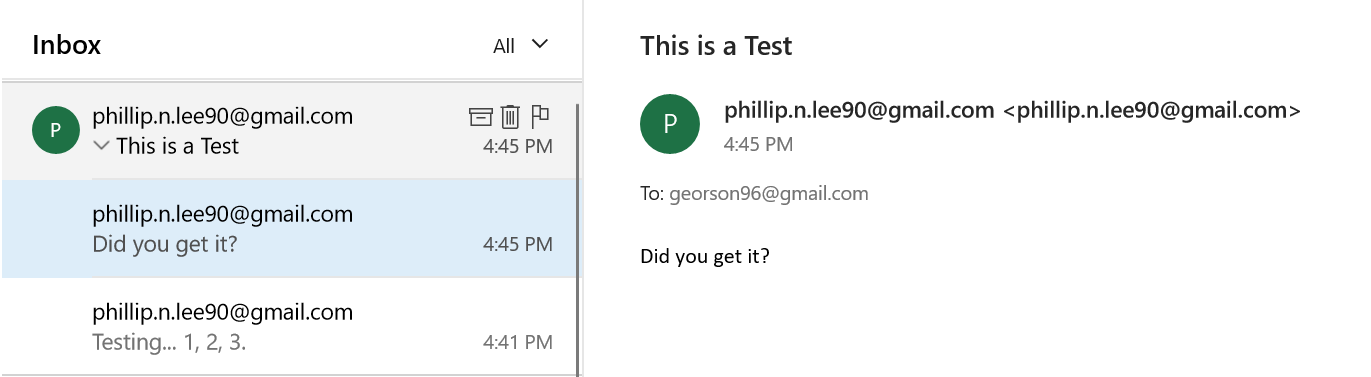


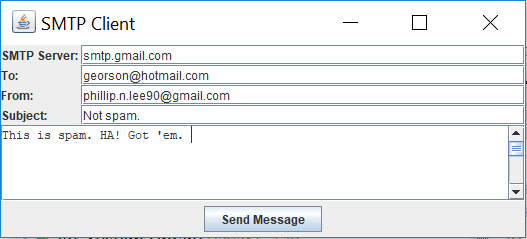
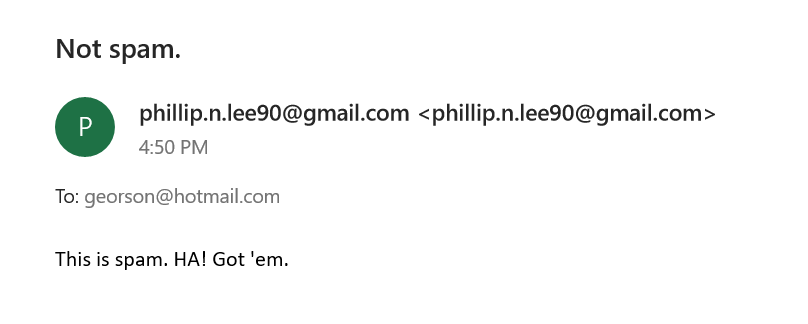
**Part II**

In Part II of the experiment, I set out to run the SMTPClient.java program, which will send an email from an application.  
  
**Steps 1-3:** I created a program utilizing the Example 19.2 (SMTPClient.java) code from the Java Network Programming textbook provided by my professor. Just like with Part I, I modified the SMTPClient.java code so that my program would be able to function with smtp.gmail.com as my SMTP server, thus, allowing an email to successfully be sent out. Unlike Part I, this program sends an email from an application represented by a GUI (Graphical User Interface) where the user can type in all of the email information (SMTP Server, email addresses, Subject, and message), send the email, and then can change any of the email information without rerunning the program.

Below is the result of running my program several times with variation, such as each email having different messages, one email having a different subject from the others, and one email being sent to My Hotmail address instead of his Gmail address.

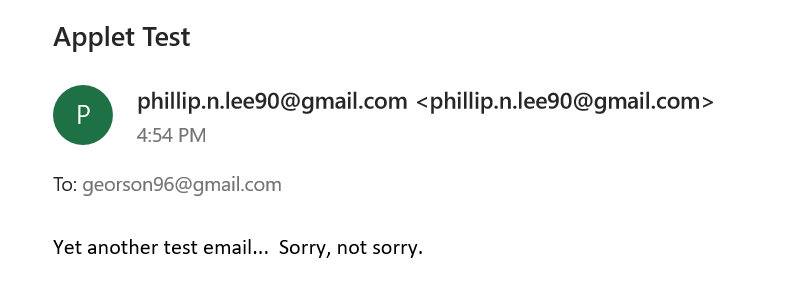
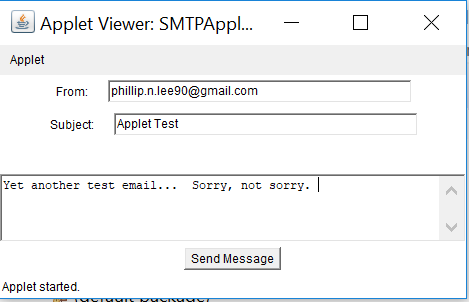
 

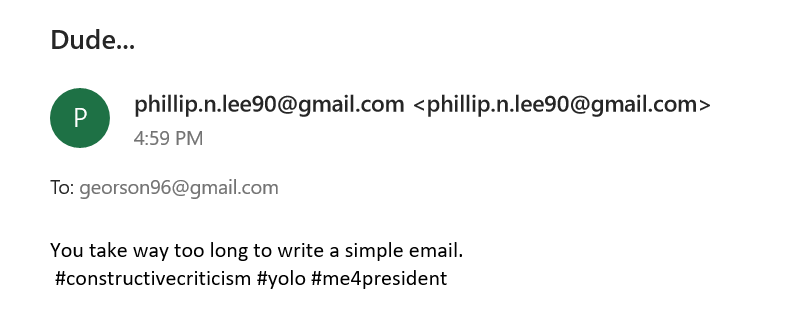
 

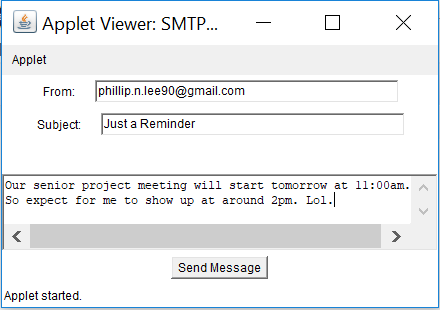
**Part III**

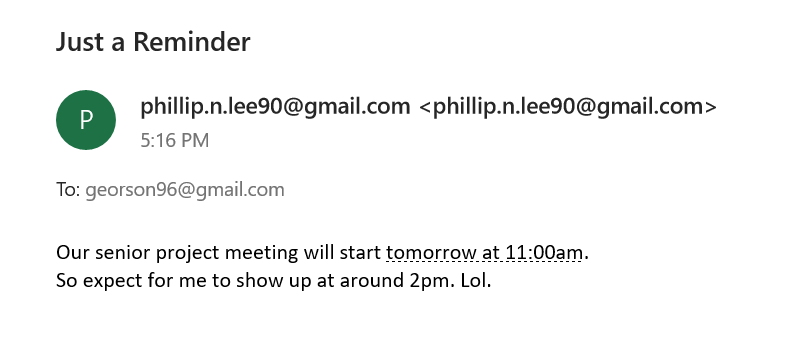
In Part III of the experiment, I set out to run the SMTPApplet.java program, which will send an email from an applet.  
  
**Steps 1 & 2:** I created a program utilizing the SMTPApplet.java code provided by my professor. Just like with Parts I and II, I modified the SMTPApplet.java code so that my program would be able to function with smtp.gmail.com as my SMTP server, thus, allowing an email to successfully be sent out. Similar to how Part II sent an email from an application, this program sends an email from an applet represented by a GUI where the user can type in some of the email information (“From” email address, Subject, and message), send the email, and then can change that email information without rerunning the program. However, if the user wants to change the SMTP server or the “To” email address, then they will have to change the code and then rerun the program.

Below is the result of running my program several times with variation, such as each email having different messages and different subjects.







**Conclusion**

This project helped give me a basic understanding of how the email-sending process operates. Also, this project demonstrated how a computer program can be used to generate and send out an email, thus, explaining how websites, such as Amazon, are able to send automated emails. I learned the importance of an SMTP (Simple Mail Transfer Protocol) server, in which it is the machine that handles the sending of emails. Another thing that I discovered from completing this project is how using an application or applet (like in Parts II and III) to send emails was a lot easier than having all of the email information hard-coded (like in Part I). By using an applet or especially by using an application, I could quickly change the information (such as the message or subject) that was displayed in a very user-friendly way on a GUI (Graphical User Interface) while not having to rerun the program.

The biggest roadblock in my efforts to get my code to function properly ended up being Cal Poly Pomona’s SMTP server. I initially tried using the college’s SMTP server (exchange.cpp.edu) to test out Part I, but I found that the program was not able to send out any emails. I then skipped ahead to Parts II and III using the college’s SMTP server and ran into the same problem where no emails were being sent out. I eventually realized that the college’s SMTP server was preventing any emails from being sent out from my programs, most likely in an effort to avoid spam/phishing emails. For Part I, I decided to switch to using Google’s SMTP server (smtp.gmail.com) and used a classmate’s Gmail address for the “from” address, which required revising the code. After making Part I’s code compatible with Google’s SMTP server (which included putting the Gmail address and Gmail password into strings that would be placed into a function so that the server could authenticate the Gmail address and allow for emails to be sent from it), I ran the program and found that it successfully sent out an email. After getting Part I to work properly, I was quickly able to get Parts II and III to successfully send out emails as well.