

## Mail Client

The goal of the assignment is to design and implement a mail client that sends email to any recipient.

Your program — call it `mymail.py` — should work as follows; after you give the command:

```
mymail.py somebody@someserver.com some.smtp.server
```

it allows the user to enter text, until you write a single period on a new line, and press `cr` and `lf`. For example:

```
~$ mymail.py soltys@mcmaster.ca ritchie.cas.mcmaster.ca
Hello,
How are you doing?
Bob
.
```

Your program will then send the message you just typed to the email `soltys@mcmaster.ca` (or any other address that you used) using the SMTP server `ritchie.cas.mcmaster.ca` (or any other SMTP server that you chose). On the next page you are given a “skeleton Python code,” as an example. Note that the skeleton assumes that the message is given in the variable `msg`, and the SMTP server is given in the variable `mailserver`, but your program will accept the a typed-in message instead, and use the SMTP server given in the command line.

Also, Python provides a module, called `smtplib`, which has built in methods to send mail using SMTP protocol. However, you will **not** be using this module in this lab, because it hides the details of SMTP and socket programming.

Once you have the above working, make the following two additions:

1. Add a switch `-s` for “Secure Connections,” that is,

```
mymail.py -s somebody@someserver.com smtp.gmail.com
```

Mail servers like Google mail (address: `smtp.gmail.com` and `port: 587`) require your client to add a Transport Layer Security (TLS) or Secure Sockets Layer (SSL) for authentication and security reasons, before you send `MAIL FROM` command. When the switch `-s` is present, add TLS/SSL commands to your existing ones and implement your client using Google mail server at above address and port.

2. Discuss, but do not implement, how would you handle attachments.

## Skeleton Python Code for the Mail Client

```
from socket import *
msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"
# Choose a mail server, e.g., ritchie.cas.mcmaster.ca, and call it mailserver
mailserver = #Fill in start ... #Fill in end
# Create socket called clientSocket and establish a TCP connection
# with mailserver
#Fill in start
...
#Fill in end

recv = clientSocket.recv(1024)
print recv
if recv[:3] != '220':
    print '220 reply not received from server.'
# Send HELO command and print server response.
heloCommand = 'HELO Alice\r\n'
clientSocket.send(heloCommand)
recv1 = clientSocket.recv(1024)
print recv1
if recv1[:3] != '250':
    print '250 reply not received from server.'
# Send MAIL FROM command and print server response.
# Fill in start
...
# Fill in end

# Send RCPT TO command and print server response.
# Fill in start
...
# Fill in end

# Send DATA command and print server response.
# Fill in start
...
# Fill in end

# Send message data.
# Fill in start
...
# Fill in end

# Message ends with a single period.
# Fill in start
...
# Fill in end

# Send QUIT command and get server response.
# Fill in start
...
# Fill in end
```