CSUS, College of Engineering and Computer Science
Department of Computer Science
CSC 138 – Computer Network and Internet

Socket Programming Assignment 3 – Mail Client
Due Date: 11/12/2017, 11:59pm

Goal: Socket programming assignments are to help you review and apply your conceptual knowledge from this class.

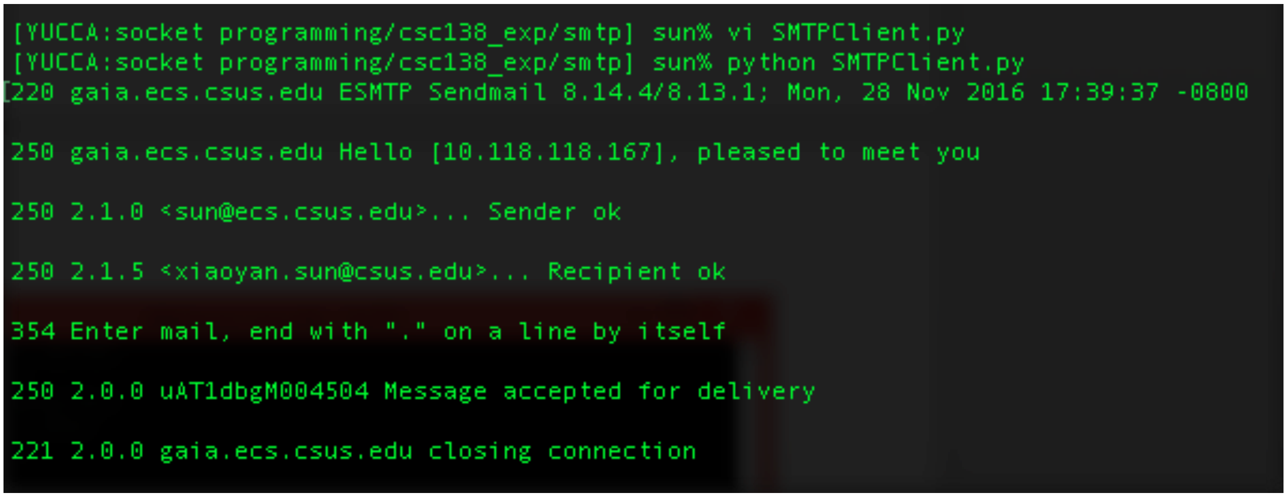
Attention: Although the examples discussed in class are in Python/C, your submission can be in Python/C/Java. If you choose do so, the caveat is that there is more help (see below) if you do it in Python. Code plagiarism is absolutely **NOT** allowed! Please prepare for a **demonstration** of running your program in front of the instructor/grader and answer their questions.

Instructions: (Textbook Page 180) This is the second of a series of programming assignments in the text book that will be assigned in the course of the semester. Students can find full details of these assignments, as well as important snippets of the Python code, at the Web site for the text book. http://wps.pearsoned.com/ecs_kurose_compnetw_6/.

The goal of this programming assignment is to create a simple mail **client** that sends email to any recipient. Your client will need to establish a TCP connection with a mail server (e.g. ECS mail server: gaia.ecs.cus.edu, or the SMTP server of CSUS: smtp.csus.edu), dialogue with the mail server using the SMTP protocol, send an email message to a recipient (e.g., your friend or yourself) via the mail server, and finally close the TCP connection with the mail server. Python provides a module, called `smtplib`, which has built in methods to send mail using SMTP protocol. However, we will not be using this module in this lab, because it hides the details of SMTP and socket programming.

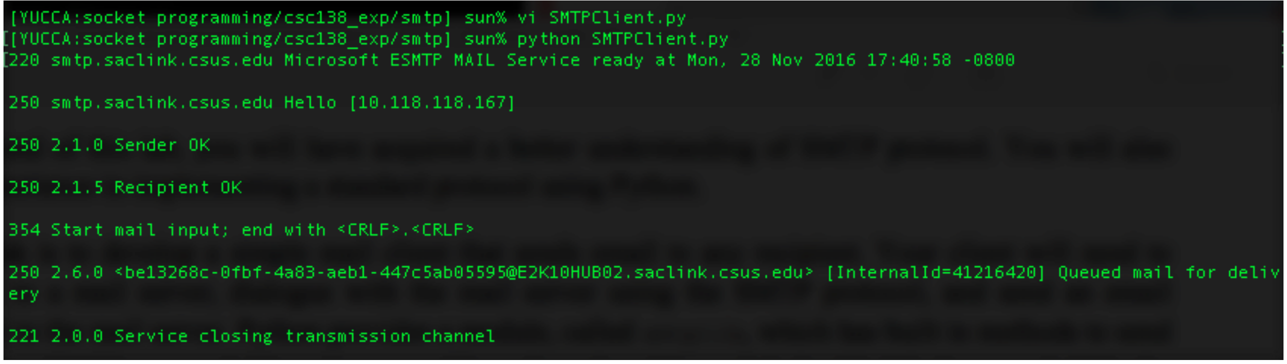
For this assignment, the **textbook**'s **companion Web site** provides the skeleton code for your client. Your job is to complete the code and test your client by sending email to different user accounts. You may also try sending through different servers (for example, through ECS mail server: gaia.ecs.csus.edu).

If you are using the mail server gaia.ecs.csus.edu, the screenshot for code execution would look like:



```
[YUCCA:socket programming/csc138_exp/smtp] sun% vi SMTPClient.py
[YUCCA:socket programming/csc138_exp/smtp] sun% python SMTPClient.py
[220 gaia.ecs.csus.edu ESMTP Sendmail 8.14.4/8.13.1; Mon, 28 Nov 2016 17:39:37 -0800

250 gaia.ecs.csus.edu Hello [10.118.118.167], pleased to meet you

250 2.1.0 <sun@ecs.csus.edu>... Sender ok

250 2.1.5 <xiaoyan.sun@csus.edu>... Recipient ok

354 Enter mail, end with "." on a line by itself

250 2.0.0 uAT1dbgM004504 Message accepted for delivery

221 2.0.0 gaia.ecs.csus.edu closing connection
```

If you are using the mail server smtp.csus.edu, the screenshot for code execution would look like:



```
[YUCCA:socket programming/csc138_exp/smtp] sun% vi SMTPClient.py
[[YUCCA:socket programming/csc138_exp/smtp] sun% python SMTPClient.py
[220 smtp.saclink.csus.edu Microsoft ESMTP MAIL Service ready at Mon, 28 Nov 2016 17:40:58 -0800

250 smtp.saclink.csus.edu Hello [10.118.118.167]

250 2.1.0 Sender OK

250 2.1.5 Recipient OK

354 Start mail input; end with <CRLF>.<CRLF>

250 2.6.0 <be13268c-0fbf-4a83-aeb1-447c5ab05595@E2K10HUB02.saclink.csus.edu> [InternalId=41216420] Queued mail for deliv
ery

221 2.0.0 Service closing transmission channel
```

Screenshot of SMTPClient

Deliverable:

An **electronic submission** of lab report should be submitted to my SacCT before the deadline. You should include both your **source code** (could be a source file or a screenshot) and at least **two screenshots** that can help you demonstrate your work: *one screenshot for the execution of source code in terminal, one screenshot for displaying the email in your email agent, such as outlook or a web browser*. Otherwise, penalty will be given in grading. Code plagiarism is absolutely **NOT** allowed! Please also prepare for a **demonstration** of running your program in front of the instructor/grader and answer their **questions** (which are about your code). You grade will be based on both the report and your performance during demonstration (**only upon request**).

Requirement: The report will all be evaluated based on the following grading criteria.

Report Correctness, Completeness, Clarity	20%+15%+15%
Demonstration Correctness, Completeness, Question	20%+15%+15%