The first material I looked at was Oracle's tutorials: http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/javamail/javamail.html.

However, I opted to follow this tutorial instead: https://www.tutorialspoint.com/javamail_api/index.htm as they ran their examples from the command line. I noticed that they used the activation.jar and javax.mail.jar files in their program. I got each from Oracle's site here: http://www.oracle.com/technetwork/java/index-138643.html

I have never run .jar files in the classpath before, so I had to look up how to compile and run it. Although the tutorial had it, it was unknowingly to me in linux format (with colons instead of semicolons). Looking up some basic information here: https://stackoverflow.com/questions/9395207/how-to-include-jar-files-with-java-file-and-compile-in-command-prompt I knew how to compile and run it in windows. What I did was include the .jar files in the same folder where my program was. Then I compile and ran the programs so I wouldn't have to include the path for them.

As I was working through the tutorial on tutorialspoint.com, I noticed each example program included a package com/tutorialspoint. I got rid of it so I wouldn't have to include the directories com and tutorialspoint. As mentioned in the Reflective Component, I decided to use multipart for Question1&2 so that I could send an attachment without overwriting the main text of the email. At first, I thought it had to do with the type of server I was using. Some sources (on StackOverflow) mentioned that pop3 would do that and I had to use imap. However, the problem was still consistent until I used multipart. All in all, I didn't have too many problems. JavaMail, I find, is a well-made API and well documented.