SSC (Communication and networking) Assignment 1: email client (4% of your finally mark)

Deadline	You must have submitted your code to Canvas for this exercise by Thursday 12th November 2015, at 9:00. No submissions will be accepted after this date, except in cases where we have explicit permission to grant an extension from the welfare team - Students with extensions granted should also notify the course lecturer by email. Vivas will take place in the lab sessions on Thursday 12th Nov. 10am-14pm, Friday 13th Nov. 13pm-17pm, and Monday 16th 11:00-14:00. Vivas will be scheduled by a timetable, which will be published on the module web page. The viva must demonstrate your code working on the lab machines.
Marking scheme	This exercise is worth 4% of your total mark for SSC. Your mark will be determined by the amount of the exercise you have completed and the quality of your solution.
Marking format	Marks will be awarded by viva and may be altered by supplementary tests, including plagiarism detection, which we perform on your electronic submission. You must submit your code electronically before your viva. You should also fill out a viva form before your viva so as not to delay the demonstration. The code that you demonstrate in your viva must be the code that you submitted electronically.

Introduction

In this exercise, you are required to implement an email client based on IMAP (for receiving email) and SMTP (for sending email) using JavaMail API

Requirements:

1. Implement the following features:

   1.1 Getting subjects of all emails from inbox and display them. The user can select a subject to display the email content (only text/plain type).

   1.2 Displaying and setting email flags appropriately. For example, if an email has been read (Flags.Flag.SEEN), your email client can display the flag appropriately. Your

email client should at least be able to display and set whether an email has been read or not.

1.3 A simple editor for inputting email address (including cc), email body and subject. The user can also select files as attachments in the editor.

1.4 Sending emails with attachments.

1.5 Searching email that contain the specified string in the header or body of the message (See http://www.faqs.org/rfcs/rfc2060.html)

1.6 A simple rule-based email filter to set custom flags to emails based on some keywords set by the user. For example, the user can set the flag of an email as 'spam' if its body contains "lucky winner". You email client should be able to display these custom flags.

1.7 A Graphic User Interface (GUI) for all the features (1.1-1.7) listed above.

2. The code you will write should be as readable as possible, e.g., informative and concise comments, etc.

Marking scheme:

Readability (total 30%):

15% for quality of code, e.g., program is generally readable and well organised

15% Informative and concise comments

Function (total 70%):

70% for implementing all the functions: 10% each for features 1.1-1.7.

Some further notes:

1. You can use Eclipse WindowBuilder or NeatBean IDE GUI buider to build you GUI. Here is a few tutorials:

   http://www.vogella.com/articles/EclipseWindowBuilder/article.html

   https://developers.google.com/java-dev-tools/wbpro/

   https://netbeans.org/kb/docs/java/quickstart-gui.html

2. You can use Gmail and you are free to choose any other email accounts.