

Summary:

THIS ASSIGNMENT IS GRADED. DO NOT COPY – THIS IS A TURNITIN ASSIGNMENT AND CODE PLAGIARISM WILL BE CHECKED AUTOMATICALLY. There will be a viva for this assignment.

In this assignment, you will be implementing a command-line based email client.

The email client has two types of recipients, official and personal. Some official recipients are close friends.

Details of the recipient list should be stored in a text file. An official recipient's record in the text file has the following format: official: <name>, <email>,<designation>. A sample record for official recipients in the text file looks as follows:

Official: nimal,nimal@gmail.com,ceo

A sample record for official friends in the text file looks as follows (last value is the recipient's birthday):

Office_friend: kamal,kamal@gmail.com,clerk,2000/12/12

A sample record for personal recipients in the text file looks as follows (last value is the recipient's birthday):

Personal: sunil,<nick-name>,sunil@gmail.com,2000/10/10

The user should be given the option to update this text file, i.e. the user should be able to add a new recipient through command-line, and these details should be added to the text file. [file handling will be covered in the 11th week]

When the email client is running, an object for each email recipient should be maintained in the application. For this, you will have to load the recipient details from the text file into the application. For each recipient having a birthday, a birthday greeting should be sent on the correct day. Official friends and personal recipients should be sent different messages (e.g. Wish you a Happy Birthday. <your name> for an office friend, and hugs and love on your birthday. <your name> for personal recipients). But all personal recipients receive the same message, and all office friends should receive the same message. A list of recipients to whom a birthday greeting should be sent is maintained in the application, when it is running. When the email client is started, it should traverse this list, and send a greeting email to anyone having their birthday on that day.

The system should be able to keep a count of the recipient objects. Use static members to keep this count.

All the emails sent out by the email client should be saved into the hard disk, in the form of objects – object serialization can be used for this. The user should be able to retrieve information of all the mails sent on a particular day by using a command-line option. [object serialization will be covered in the 11th week]

You only have to send out messages. No need to implement the logic to receive messages.

Command-line options should be available for:

Adding a new recipient

Sending an email

Printing out all the names of recipients who have their birthday set to current date

Printing out details (subject and recipient) of all the emails sent on a date specified by user input

Printing out the number of recipient objects in the application

You may use the code given in this link to implement the basic functionality of the mail client (But it is perfectly ok to use a different code as well). In the given code, note that it imports the javax.mail package. This package is included in the javax.mail.jar, which can be downloaded from here.

You are given marks for the

Correct implementation of the mail sending functions (i.e. sending a birthday greeting, sending an email based on the instructions given through command-line, ability to serialize email objects, etc).

Correct use of OOP principles

Use of coding best practices

Use the given Email_Client.java file as the starting point.

Save the recipient data into clientList.txt.

Submission information: Put all the code into one text file, convert it to a pdf, and submit.