```
package Email_Client;
// Index no : 200407H

//import libraries

import java.util.Scanner;

class Email_Client {

    public static void main(String[] args) {

        // start email client
        // code to create objects for each recipient in clientList.txt


        // creating the clientList.txt if doesn't exist
        CreateFile clientList = CreateFile.createfile();

        //creating objects for each recipient in clientList.txt
        ReadFile.readFile(clientList.getfilePath());



        // sending birth wishes

        // whenever new friend object is created it will be sent to the Birthday class
        // if the friends birthday is today it will send a birthday wish



        Scanner scanner = new Scanner(System.in);
        boolean exit = false;
        // taking inputs from the user
        while(!exit){
            System.out.println("Enter option type: \n"
                    + "1 - Adding a new recipient\n"
                    + "2 - Sending an email\n"
                    + "3 - Printing out all the recipients who have birthdays\n"
                    + "4 - Printing out details of all the emails sent\n"
                    + "5 - Printing out the number of recipient objects in the application");

            int option = scanner.nextInt();
            scanner.nextLine(); // to throw away the \n in the buffer of the scanner object

            switch(option){


                case 1:
                    System.out.println("Add recipient details");
                    // input format - Official: nimal,nimal@gmail.com,ceo
                    while(!scanner.hasNext());
                    String new_recipient =" " ;
                    if (scanner.hasNext()) {

                        // get details of the recipient using a single input
```

```
        new_recipient = scanner.nextLine();

    }

    // get the filePath of clientList.txt file
    String filePath = clientList.getfilePath();
    // store details in clientList.txt file
    WriteToFile.addRecipient(filePath, new_recipient);

    // code to add a new recipient
    ReceipientConstructor.constructor(new_recipient);
    // new recipient obj will be created in one of the following forms
    // Official
    // Official_friend
    // Personal

    break;


case 2:
// sending an email

    System.out.println("Enter the details: email, subject, content ");
    // input format - email, subject, content
    while(!scanner.hasNext());
    String input =" " ;
    if (scanner.hasNext()) {

        // get details of the recipient using a single input
        input = scanner.nextLine();

    }

    String [] arr = input.split(", ",3);
    // creating an email obj
    Email email = new Email(arr[0],arr[1],arr[2]);
    //sending the email
    emailSender.sendEmail(email);

    break;




case 3:
    System.out.println("Enter the Birthday ");
    // input format - yyyy/MM/dd (ex: 2018/09/17)
    while(!scanner.hasNext());
    String birthday =" " ;
    if (scanner.hasNext()) {

        // get details of the recipient using a single input
        birthday = scanner.nextLine();

    }
```

```
                // code to print recipients who have birthdays on the given date
                Recipients.listOfBirthdayPeople(birthday);

                break;




            case 4:
                System.out.println("Enter the date in the format, yyyy_MM_dd: ");
                // input format - yyyy_MM_dd (ex: 2018_09_17)

                while(!scanner.hasNext());
                String date =" " ;
                if (scanner.hasNext()) {

                    date = scanner.nextLine();
                }

                // calling methods to print the details
                Email.getDetails(date);
                break;




            case 5:
                // code to print the number of recipient objects in the application
                System.out.println(ReceipientConstructor.getnumOfRecipients());

                break;



        }
        System.out.println ("Want to exit the email client : Yes / No ");
        while(!scanner.hasNext());
        String next =" " ;
        if (scanner.hasNext()) {

            next = scanner.nextLine();
        }

        if(next.equals("Yes")){
            exit = true;
        }

    }
    scanner.close();

  }
}
```

==========

```
package Email_Client;

class ReceipientConstructor{

   private static int numOfRecipients=0;

   public static void constructor(String new_record){

      numOfRecipients+=1;

      String [] arr = new_record.split(" ",2);
      System.out.println(arr[0]);

      if(arr[0].equals("Official:")){

         String [] record = arr[1].split(",",3);
         // creating new recipient as a Official obj
         new Official(record[0], record[1],record[2]);
      }


      else if (arr[0].equals("Office_friend:")){


         String [] record = arr[1].split(",",4);
         // creating new recipient as a Official_friends obj
         new Officials_friends(record[0], record[1],record[2],record[3]);

      }


      else if(arr[0].equals("Personal:")){

         String [] record = arr[1].split(",",4);
         // creating new recipient as a Personal obj
         new Personal(record[0], record[1],record[2],record[3]);

      }

   }


   // function to return number of Recipient objects at the moment
   public static int getnumOfRecipients(){

      return numOfRecipients;
   }

}
```

==========

```
package Email_Client;

import java.util.ArrayList; // import ArrayList class
```

```
abstract class Recipients {
   // defining common attributes for all the recipients
   //name, email address

   private String name ;
   private String email;

   // static variable to store the recipients to whom a birthday greeting should be sent
   public static ArrayList<Friends> ListOfBDays = new ArrayList<Friends>();

   public Recipients(String name , String email){

      // create recipient obj
      this.name = name;
      this.email = email;
   }


   // defining methods common for all recipients

   // returning the name of the obj

   public String getname(){
      return this.name;
   }

   // returning the email of the obj
   public String getemailaddress(){
      return this.email;
   }



   public static ArrayList<Friends> getListOfBDays(){
      // returns the list of recipientsa to whom a birthdat greeting should be sent
      return ListOfBDays;
   }


   //method to print the list of people who has birthdays in the given date
   public static void listOfBirthdayPeople(String date){

      for(int i=0; i< ListOfBDays.size(); i++){

         String s =(ListOfBDays.get(i)).getbirthday();

         if(s.equals(date)){

            System.out.println((ListOfBDays.get(i)).getname());
         }

      }
```

```
    }

}
```

```
package Email_Client;

class Official extends Recipients {
    // sub class of recipient class

    // defining attributes special for Official objects

    private String designation ;

    public Official(String name , String email, String designation){
        // create official objects

        super( name, email);

        this.designation=designation;
    }

}
```

```
package Email_Client;

class Officials_friends extends Official implements Friends  {

    // defining atrributes special for Official_friends

    private String birthday;


    public Officials_friends(String name, String email, String designation, String birthday){
        // create official_friends obj

        super(name, email, designation);
        this.birthday=birthday;

        // adding the object to the ListOfBDays arrayList
        ListOfBDays.add(this);
        Birthday.birthdayPeople(this);
    }


    // return the Birth date
    public String getbirthday(){
        return this.birthday;

    }

    public void sendBirthdayWishes(){
```

```
        // constructing the email obj
        Email email = new Email(this.getemailaddress(), "Birthday Wishes", "Wish you a Happy Birthday.\nNuthara.")
;

        // calling for emailsender to send the birthday wishes
        emailSender.sendEmail(email);

    }



}
```

==========

```
package Email_Client;

class Personal extends Recipients implements Friends{

    //defining aattributes special for personal objects

    private String nickName;
    private String birthday;

    public Personal(String name, String nickName , String email, String birthday){
        // create Personal obj

        super(name, email);
        this.birthday=birthday;
        this.nickName=nickName;

        // adding the object to the ListOfBDays arrayList
        ListOfBDays.add(this);
        Birthday.birthdayPeople(this);
    }

    // return the Birth date
    public String getbirthday(){
        return this.birthday;

    }
    public void sendBirthdayWishes(){

        // constructing the email obj
        Email email = new Email(this.getemailaddress(), "Birthday Wishes", "Hugs and love on your birthday.\nNuthar
a.");

        // calling for emailsender to send the birthday wishes
        emailSender.sendEmail(email);
    }
}
```

==========

```
package Email_Client;

interface Friends {
   //defining common methods to friends objects
   public String getname();
   public String getbirthday();
   public void sendBirthdayWishes();

}
```

==========

```
package Email_Client;

import java.time.LocalDate; // import LocalDate class
import java.time.format.DateTimeFormatter; // import DateTimeFormatter class


class Birthday {

   //method to select those who have birthday today out of all the friends in the given list and call the method in relev
ant class to send the birthday wishes
   public static void birthdayPeople(Friends friend){

      DateTimeFormatter formatter = DateTimeFormatter.ofPattern("MM/dd");
      String date = formatter.format(LocalDate.now());

      if(((friend.getbirthday()).substring(5)).equals(date)){

         friend.sendBirthdayWishes();
      }

   }
}
```

==========

```
package Email_Client;

import java.io.File;  // Import the File class
import java.io.IOException;  // Import the IOException class to handle errors


class CreateFile{
   // this class should have only one instance

   private static CreateFile clientList;
   private String filePath;

   private CreateFile(){

      filePath ="D:\\Nuthara_N_R\\UOM\\Academic\\Sem 2\\CS1040 - Program Construction\\Practical assessments
\\Email_client\\clientList.txt";
```

```
    }

    public static CreateFile createfile(){

        if(clientList==null){

            clientList= new CreateFile(); // a new create file obj will be created at every initialization of Email_Client

            try {
                File clienttxt = new File(clientList.filePath);
                clienttxt.createNewFile(); // create a file if there is no file in the given filepath , else doesn't create a file
              } catch (IOException e) {
                System.out.println("An error occurred.");
                e.printStackTrace();
              }


        }
        return clientList;

    }

    public String  getfilePath(){
        // returning the filePath of clientList.txt
        return this.filePath;

    }

}
```

==========

```
package Email_Client;

import java.io.File;  // Import the File class
import java.io.FileNotFoundException;  // Import this class to handle errors
import java.util.Scanner; // Import the Scanner class to read text files

class ReadFile {

    public static void readFile(String filePath ){

        try {
            File clientlist = new File(filePath);
            Scanner scanner = new Scanner(clientlist);
            while (scanner.hasNextLine()) {
              String data = scanner.nextLine();
              // constructing objts for each recipiennt in the clientlist
              ReceipientConstructor.constructor(data);
            }
            scanner.close();
          } catch (FileNotFoundException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
```

```
        }


    }

}
```

==========

```
package Email_Client;


import java.io.FileWriter;   // Import the FileWriter class
import java.io.IOException;  // Import the IOException class to handle errors
import java.io.BufferedWriter;// import the BufferWriter class

class WriteToFile {

    public static void addRecipient(String filePath , String recipient_record){

        try {
            FileWriter new_record = new FileWriter(filePath,true);
            BufferedWriter buffer= new BufferedWriter(new_record);
            buffer.write(recipient_record);
            buffer.close();

        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }

}
```

==========

```
package Email_Client;

import java.io.*; //import io class for object serialization and deserialization
import java.time.LocalDate; // import LocalDate class
import java.time.LocalTime; //import LocalDate class
import java.time.format.DateTimeFormatter; // import DateTimeFormatter class



class Email implements Serializable {

    private String to;
    private String from ="nnrandrahennedi.20@uom.lk";
    private String subject ;
    private String content;


    // constructor
    public Email(String to,  String subject, String content){
```

```
        this.to = to;
        this.subject =subject;
        this.content=content;

        // serialization of the obj
        this.emailSerialization();
    }

    // return sender of the given email
    public String getSender(){
        return this.from;
    }

    // return srecipient of the given email
    public String getRecipient(){
        return this.to;
    }

    // return subject of the given email
    public String getSubject(){
        return this.subject;
    }

    // return the content of the given email
    public String getContent(){
        return this.content;
    }



    // Email obj serializing method

    private void emailSerialization(){


        // getting the current date in the format of yyyy/mm/dd
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy_MM_dd");
        String date = formatter.format(LocalDate.now());

        // getting the time in hhmmss format
        DateTimeFormatter format = DateTimeFormatter.ofPattern("HH:mm:ss");
        String time = format.format(LocalTime.now());
        time= time.replace(":","");


        // a new derectory will be created for every day
        // each email obj will be serialized under the date of sending
        // a .ser file will be created for each email obj
        //file name:-   time_when_sending.ser


        // file path for a directory
        String filepath ="D:\\Nuthara_N_R\\UOM\\Academic\\Sem 2\\CS1040 - Program Construction\\Practical asses
sments\\Email_client\\Sent Emails\\"+date ;
```

```
        // creating a directory if doesn't exist
        File file_ = new File(filepath);
        file_.mkdir();

        // updating file path to serialization of obj
        // ex : "D:\\Nuthara_N_R\\UOM\\Academic\\Sem 2\\CS1040 - Program Construction\\Practical assessments\\E
mail_client\\Sent Emails\\2022_07_20\\152045.ser"
        filepath =filepath+"\\"+ time+".ser";

        //serialization of email obj
        try
        {
            //Saving of object in a file

            // creating a file .ser file with the file name time_when_sending.ser
            FileOutputStream file = new FileOutputStream(filepath);
            ObjectOutputStream out = new ObjectOutputStream(file);

            // Method for serialization of object
            out.writeObject(this);

            out.close();
            file.close();
        }

        catch(IOException ex)
        {
            System.out.println("IOException is caught");
        }
    }



    // method to deserialization
    //returns an Email obj

    private static Email emailDeserialization(String objPath){

        Email email;
        // Deserialization
        try
        {
            // Reading the object from a file
            FileInputStream file = new FileInputStream(objPath);
            ObjectInputStream in = new ObjectInputStream(file);
            //Email e ;
            // Method for deserialization of object

            email =(Email)in.readObject();


            in.close();
            file.close();
```

```
            return email;
        }

        catch(IOException ex)
        {
            System.out.println("IOException is caught");
        }

        catch(ClassNotFoundException ex)
        {
            System.out.println("ClassNotFoundException is caught");
        }

        return email=null;

    }



    public static void getDetails(String date){

        // calling for the obj deserialization function and returns the subject and the recipient of the emails at a given sp
ecific date

        // creating the directory path for the directory related to the given date
        String path = "D:\\Nuthara_N_R\\UOM\\Academic\\Sem 2\\CS1040 - Program Construction\\Practical assessm
ents\\Email_client\\Sent Emails\\"+date;
        //getting the directory related to the given date
        File f= new File(path);
        //populate the array with the names of all the .ser files i.e. all the Email objts
        String [] objList = f.list();

        // deserialize each obj
        for (String name: objList ){

            String objPath = path+"\\"+name;
            // passing for deserialization
            Email email = Email.emailDeserialization(objPath);
            if (email!=null){
                // printitng the email details
                System.out.println("Recipient: "+email.getRecipient());
                System.out.println("Subject: " +email.getSubject());
                System.out.println();

            }
        }
    }

}
```

==========

```
package Email_Client;

import javax.mail.*;
```

```
import java.util.*;
import javax.mail.internet.*;
import javax.mail.Message;


class emailSender {

   public static void sendEmail(Email email){

      //code to set the server senttings and authentication process
      Properties prop = new Properties();
      prop.put("mail.smtp.auth", true);
      prop.put("mail.smtp.starttls.enable", true);
      prop.put("mail.smtp.host", "submit.uom.lk");
      prop.put("mail.smtp.port", "587");



      Session session = Session.getInstance(prop, new Authenticator() {
      protected PasswordAuthentication getPasswordAuthentication() {
         return new PasswordAuthentication("nnrandrahennedi.20@uom.lk", "<UOM PASSWORD>");
      }
      });


      // creating the message and initializing the details for sending the email
      Message message = new MimeMessage(session);
      try{
      InternetAddress internetAddress = new InternetAddress("nnrandrahennedi.20@uom.lk");
      message.setFrom(internetAddress);
      message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(email.getRecipient()));

      // creating the email body
      message.setSubject(email.getSubject());

      String content = email.getContent();

      MimeBodyPart emailBody= new MimeBodyPart();
      emailBody.setContent(content, "text/plain");

      Multipart multipart = new MimeMultipart();
      multipart.addBodyPart(emailBody);

      message.setContent(multipart);

      // sending the email
      Transport.send(message);
      System.out.println("Email sent successfully");
      }

      catch(AddressException ae){
      System.out.println("Error" + ae);

      }
      catch(MessagingException me){
```

```
        System.out.println("Error"+ me);

        }



    }


}
```
==========
==========