```
// your index number
// 200081B
// E.A.C. CHANDEEPA


//import libraries
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;



class Email_Client {

 public static void main(String[] args) throws ParseException {

    RecipientFactory RFactory = new RecipientFactory();

    ArrayList<Email> SentEmails = Serialization.deserializeEmail();


    ArrayList<Recipient> Recipients;
    Recipients = RecipientLoder.loadRecipient(RFactory);


    ArrayList<Wishable> BithdayRecipients;
    BithdayRecipients = RFactory.getBirthdayRecipientList();

    for(Wishable BRC : BithdayRecipients ){
       Email em = BRC.wishOnBirthday();
       if(em != null){
           SentEmails.add(em);
       }
    }



    boolean status = true;
    while(status){

        Scanner scanner = new Scanner(System.in);

        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\n"
        + "99 - Exit");

        int option = scanner.nextInt();

        switch(option){
        case 1:
        // input format - Official: nimal,nimal@gmail.com,ceo
        // Use a single input to get all the details of a recipient
        // code to add a new recipient
        // store details in clientList.txt file
        // Hint: use methods for reading and writing files

            System.out.println("Enter Recipient details");

            Scanner scanner2 = new Scanner(System.in);
            String recipientdata = scanner2.nextLine();

            Recipient rc = RFactory.getRecipient(recipientdata);
            Recipients.add(rc);

            if(rc!=null){
            FileHandling myFile = new FileHandling();
            myFile.writeFile("clientList.txt", recipientdata);
            }

        break;

        case 2:
        // input format - email, subject, content
        // code to send an email

            System.out.println("Enter email details");
            Scanner scanner3 = new Scanner(System.in);
            String emailData = scanner3.nextLine();
            Email em = new Email();
            if(em.createEmail(emailData)){
            em.sendMail();
            SentEmails.add(em);
            }

        break;

        case 3:
        // input format - yyyy/MM/dd (ex: 2018/09/17)
        // code to print recipients who have birthdays on the given date

             System.out.println("Enter Birthday ");
             Scanner scanner4 = new Scanner(System.in);
             String birthday = scanner4.nextLine();
```

```
             BithdayRecipients = RFactory.getBirthdayRecipientList();

             for(Wishable BRC : BithdayRecipients ){
             if(BRC.CheckBirthday(birthday)){
                BRC.getDetails();
            }
        }



        break;

        case 4:
        // input format - yyyy/MM/dd (ex: 2018/09/17)
        // code to print the details of all the emails sent on the input date

           System.out.println("Enter Date ");
           Scanner scanner5 = new Scanner(System.in);
           String date = scanner5.nextLine();

           try {
           Date dt = new SimpleDateFormat("yyyy/MM/dd").parse(date);

           for (Email em2 : SentEmails){
               if(em2.getSendDate().compareTo(dt)==0)
               {
                   System.out.println(em2.getEmial()+ " , "+em2.getSubject());
               }
           }
           }
           catch(ParseException ex) {
                    System.out.println("Error , Please use the correct input
format!!");
               }

        break;

        case 5:
        // code to print the number of recipient objects in the application

            System.out.println("Number of Recipients : ");
            int tot = Recipient.getTotalRecipient();
            System.out.println(tot);

        break;

        case 99:
        // Exit the programm

            Serialization.Serialize(SentEmails,"SentEmail.ser");
            status = false;
            scanner.close();

        break;
        }


        System.out.println("\n \n");
        // start email client
        // code to create objects for each recipient in clientList.txt
        // use necessary variables, methods and classes

    }
  }
}
// create more classes needed for the implementation (remove the public access
modifier from classes when you submit your code





abstract class Recipient {

    //Official: nimal,nimal@gmail.com,ceo
     private String type;
     private String name ;
     private String email ;
     private static int tot_recipient;

    public Recipient(){
    }

    public Recipient(String type , String name,String email){
        this.type = type;
        this.name = name;
        this.email = email;
        tot_recipient+=1;
    }

    public String getName(){
        return name;
    }

    public void setType(String type){
        this.type = type;
    }

    public String getType(){
        return type;
    }

    public void setEmail(String email){
        this.email = email;
    }

    public String getEmail(){
        return email;
    }

    public static int getTotalRecipient()
    {
        return tot_recipient;
    }

}





class Official_recipient extends Recipient{
    private String  designation;

    public Official_recipient(){
    }

    Official_recipient(String type , String name, String Email , String 
designation){
        super(type, name,Email);
        this.designation = designation;
    }

    public void SetDesignation(String designation)
    {
        this.designation=designation ;
    }
    public String getDesignation()
    {
        return designation;
    }

}
```

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

 class Personal_recipient extends Recipient implements Wishable {
    private Date birthday;
    private String nickName;


    public Personal_recipient(String type,String name, String nickName, String email
, Date birthday ){
        super(type,name , email);
        this.birthday = birthday;
        this.nickName = nickName;
    }

    public void setBirthday(Date birthday)
    {
        this.birthday = birthday;
    }

    public Date getBirthday(){
        return birthday;
    }

    public void setNickName(String nickName)
    {
        this.nickName = nickName;
    }

    public String getNickName(){
        return nickName;
    }

    /**
     *
     * @return
     */
    @Override
    public Email wishOnBirthday()
    {
        Date now = new Date();

        Calendar cal = Calendar.getInstance();
        cal.setTime(now);
        int month = cal.get(Calendar.MONTH);
        int date = cal.get(Calendar.DAY_OF_MONTH);

        Calendar cal2 = Calendar.getInstance();
        cal2.setTime(this.birthday);
        int month2 = cal2.get(Calendar.MONTH);
        int date2 = cal2.get(Calendar.DAY_OF_MONTH);

        if(month ==month2 && date == date2){
            String Eaddress = this.getEmail();
            String Body = "hugs and love on your birthday "+this.getName();
            String Subject = "Birthday Greeting ";

            Email em = new Email(Eaddress, Subject, Body);
            em.sendMail();
            return em;
        }
        return null;
    }

    @Override
    public boolean CheckBirthday(String birthday){
        try {
            Date date;
            date = new SimpleDateFormat("yyyy/MM/dd").parse(birthday);

            Calendar cal = Calendar.getInstance();
            cal.setTime(date);
            int month1 = cal.get(Calendar.MONTH);
            int date1 = cal.get(Calendar.DAY_OF_MONTH);

            Calendar cal2 = Calendar.getInstance();
            cal2.setTime(this.birthday);
            int month2 = cal2.get(Calendar.MONTH);
            int date2 = cal2.get(Calendar.DAY_OF_MONTH);

            return month1 ==month2 && date1 == date2;

        } catch (ParseException ex) {
            return false;
        }
    }

     @Override
    public void getDetails(){
        System.out.println(this.getName());
    }
}
```

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;


class Office_friend extends Official_recipient implements Wishable{
    private Date birthday;

    public Office_friend(String type , String name , String email, String
designation , Date birthday){
        super(type , name, email, designation);
        this.birthday=birthday;
    }

    public void setBirthday(Date birthday){
        this.birthday = birthday;
    }

    public Date getBirthday(){
        return birthday;
    }

    /**
     *
     * @return
     */
    @Override
    public Email wishOnBirthday()
    {
        Date now = new Date();

        Calendar cal = Calendar.getInstance();
        cal.setTime(now);
        int month = cal.get(Calendar.MONTH);
        int date = cal.get(Calendar.DAY_OF_MONTH);

        Calendar cal2 = Calendar.getInstance();
        cal2.setTime(this.birthday);
        int month2 = cal2.get(Calendar.MONTH);
        int date2 = cal2.get(Calendar.DAY_OF_MONTH);

        if(month ==month2 && date == date2){
           String EAddress = this.getEmail();
           String Body = "Wish you a Happy Birthday "+this.getName();
           String Subject = "Birthday Greeting ";

           Email em = new Email(EAddress, Subject, Body);
           em.sendMail();
           return em;
        }
        return null;

    }

    @Override
    public boolean CheckBirthday(String birthday){
        try {
            Date date;
            date = new SimpleDateFormat("yyyy/MM/dd").parse(birthday);

            Calendar cal = Calendar.getInstance();
            cal.setTime(date);
            int month1 = cal.get(Calendar.MONTH);
            int date1 = cal.get(Calendar.DAY_OF_MONTH);

            Calendar cal2 = Calendar.getInstance();
            cal2.setTime(this.birthday);
            int month2 = cal2.get(Calendar.MONTH);
            int date2 = cal2.get(Calendar.DAY_OF_MONTH);

            return month1 ==month2 && date1 == date2;

        } catch (ParseException ex) {
            return false;
        }
    }

    @Override
    public void getDetails(){
        System.out.println(this.getName());
    }
}





interface Wishable {
    public Email wishOnBirthday();
    public boolean CheckBirthday(String date);
    public void getDetails();

}
```

```java
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;


class RecipientFactory {

    ArrayList<Wishable> BirthdayRecipients;

    public RecipientFactory(){
        BirthdayRecipients = new ArrayList<>();
    }

    public Recipient getRecipient(String recipientData){

        try{
        String str = recipientData;
        String rsData[] = str.split(" ");
        String data[] = rsData[1].split(",");


        if(rsData[0].equals("Official:")){
            Official_recipient rcs = new
Official_recipient("Official",data[0],data[1],data[2]);
            return rcs;

        }
        else if(rsData[0].equals("Office_friend:")){
            String sDate1 = data[3];
            Date date1;
            try {
                date1 = new SimpleDateFormat("yyyy/MM/dd").parse(sDate1);
                Office_friend rcs =  new
Office_friend("Office_friend",data[0],data[1],data[2],date1);
                BirthdayRecipients.add(rcs);
                return rcs;

            } catch (ParseException ex) {

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

            String sDate2 = data[3];
            Date date2;
            try {
                date2 = new SimpleDateFormat("yyyy/MM/dd").parse(sDate2);
                Personal_recipient rcs =  new
Personal_recipient("Personal",data[0],data[1],data[2],date2);
                BirthdayRecipients.add(rcs);
                return rcs;

            } catch (ParseException ex) {

            }
        }
        else{
            System.out.println("Error , Please use the correct input format!!");
            return null;
        }
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Error , Please use the correct input format!!");
        }

        return null;
    }

    public ArrayList<Wishable> getBirthdayRecipientList(){
        return BirthdayRecipients;
    }

}
```

```
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;


class RecipientLoder {
    public static ArrayList<Recipient> loadRecipient(RecipientFactory Fc)
    {
            ArrayList<Recipient> Recipients;
            Recipients = new ArrayList<>();
        try{
            File myfile = new File("clientList.txt");
                try (Scanner myReader = new Scanner(myfile)) {
                    while(myReader.hasNextLine()){
                        String data = myReader.nextLine();

                        Recipient Rc ;
                        Rc = Fc.getRecipient(data);
                        Recipients.add(Rc);
                    }
                }

        }catch(IOException e){
        }

        return Recipients;
    }
}
```

```
import java.io.IOException;
import java.io.Serializable;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;


class Email implements Serializable {
    private String Subject;
    private String email;
    private String body;
    private Date sendDate;

    public Email(){

    }

    public Email(String Email,String Subject, String body ) {
        this.Subject = Subject;
        this.body = body;
        this.email = Email;
    }

    public void setSubject(String Subject){
        this.Subject = Subject;
    }
    public void setEmial(String emial){
        this.email = emial;
    }
    public void setBody(String body){
        this.body = body;
    }
    public void setSendDate(Date sendDate){
        this.sendDate = sendDate;
    }

    public String getSubject(){
        return Subject;
    }
    public String getEmial(){
        return email;
    }
    public String getBody(){
        return body;
    }
    public Date getSendDate(){
        return sendDate;
    }

    public boolean createEmail(String str){
        try{
        String mailData[] = str.split(", ");
        this.email = mailData[0];
        this.Subject = mailData[1];
        this.body = mailData[2];
        return true;
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Error , Please use the correct input format!!");
            return false;

        }
    }

    public void sendMail()   {

        try{
        SendEmailTLS.sendMail(email, Subject, body);

        if(email!=null && Subject != null && body!=null){
        Date now = new Date();
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy/MM/dd");
        String formattedDate = simpleDateFormat.format(now);
        Date dt = simpleDateFormat.parse(formattedDate);
        this.sendDate = dt;
        }
        }
        catch( IOException| ParseException e){

        }
    }

}
```

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.text.ParseException;
import javax.mail.*;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import java.util.Properties;


class SendEmailTLS {

    public static void sendMail(String Reciver,String Subject , String Body  )
throws FileNotFoundException, IOException, ParseException {

        final String username = "chinthaniii2022@gmail.com";
        final String password = "slyifdtsemsuwjas";

        Properties prop = new Properties();

        prop.put("mail.smtp.host", "smtp.gmail.com");
        prop.put("mail.smtp.port", "587");
        prop.put("mail.smtp.auth", "true");
        prop.put("mail.smtp.starttls.enable", "true"); //TLS

        Session session = Session.getInstance(prop,
                new javax.mail.Authenticator() {
                    @Override
                    protected PasswordAuthentication getPasswordAuthentication() {
                        return new PasswordAuthentication(username, password);
                    }
                });

        try {

            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress(username));
            message.setRecipients(
                    Message.RecipientType.TO,
                    InternetAddress.parse(Reciver)
            );
            message.setSubject(Subject);
            message.setText(Body);

            Transport.send(message);
            System.out.println("Sent messages sucessfully!");


        } catch (MessagingException e) {
               System.out.println("Cannot send Email ");
        }
    }


}






import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;


class FileHandling {

    public void createFile(String fileName){
        try{
            File myObj = new File(fileName);
            if(myObj.createNewFile()){
               // System.out.println("File created "+ myObj.getName());
            }


        }catch(IOException e){
            System.out.println("Error occurred");
        }

    }

    public void writeFile(String fileName, String Message)
    {
        try{
            try (FileWriter myfile = new FileWriter(fileName,true)) {
                myfile.write(Message);
                myfile.write(System.lineSeparator());
            }

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


    public void readFile(String fileName)
    {
        try{
            File myfile = new File(fileName);
            try (Scanner myReader = new Scanner(myfile)) {
                while(myReader.hasNextLine()){
                    String data = myReader.nextLine();
                    System.out.println(data);
                }
            }
        }catch(IOException e){
            System.out.println("An error occurred.");
        }
    }

}
```

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;


abstract class Serialization {

    static void Serialize(Object obj , String FileName) {

        try{
        FileOutputStream fileStream2 = new FileOutputStream(FileName);

        try (ObjectOutputStream os = new ObjectOutputStream(fileStream2)) {
            os.writeObject(obj);
            os.close();
        }
        }
        catch(IOException e){

        }
    }
    static public Object deserialize(String FileName){



        try{
           FileInputStream fileStream = new FileInputStream(FileName);
            try (ObjectInputStream os = new ObjectInputStream(fileStream)) {
                Object obj = os.readObject();
                return obj;
            }

        }
        catch(IOException|ClassNotFoundException e){
        }
        return null;

    }

    static public ArrayList<Email> deserializeEmail(){
        ArrayList<Email> SentEmails ;
        Object obj = Serialization.deserialize("SentEmail.ser");
     if(obj == null){
         SentEmails  =new  ArrayList<>();
     }
     else{
     SentEmails = (ArrayList<Email>) obj;
     }

     return SentEmails;
    }
}
```