```
//index number.
//200452N Navinda Mansitha Perera (Perera D.N.M).
/*  Java Program that uses javax.mail library for mail sending and other associated
tasks achieved with
Serializable and FIle handling */
import java.util.ArrayList;
import java.util.Scanner;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import javax.mail.*;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import java.util.Properties;
import java.io.Serializable;
import java.io.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.ObjectOutputStream;


class Email_Client {

	public static void main(String[] args) {

		//code to handle date and time, prints a welcome message with the date
		LocalDate today_object = LocalDate.now();
		DateTimeFormatter myFormatObj =
DateTimeFormatter.ofPattern("yyyy/MM/dd");
		String date_now= today_object.format(myFormatObj);
		System.out.println("Welcome to Email Client\n"+"Today is
"+date_now.substring(0, 10));


		// taking input of the required task to perform

		Scanner scanner0 = 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");

		int option = scanner0.nextInt();



		//defining an ArrayList to hold A list of recipients to whom a birthday
greeting should be sent
		ArrayList<String> birthday_list = new ArrayList<String>();
		birthday_list.add("Personal: sunil,sunila,sunil@gmail.com,2022/08/09");
		birthday_list.add("Personal: sunil,sunila,sunil@gmail.com,2022/08/22");
		birthday_list.add("Personal: sunil,sunila,sunil@gmail.com,2022/08/14");
		birthday_list.add("Office_friend:
kamqweal,kamal@gmail.com,clerk,2000/08/14");
		int s = birthday_list.size();
		// seperate ArrayList to hold based on the recepient type
		ArrayList<Personal> pers_bday = new ArrayList<Personal>();
```

```
            ArrayList<OfficeFriend> ofp_bday= new ArrayList<OfficeFriend>();
            ArrayList<Official> off_bday = new ArrayList<Official>();



            //inserting entries in birthday_lists to specific recepients defined
above
            for(String item : birthday_list){
                String first = item.substring(0,10);
                if( first.equals("Official: ")){
                    int l = item.length();
                    String[] String_list = item.substring(10,l).split(",");
                    Official append_item = new Official(item.substring(0,7),
String_list[0], String_list[1],  String_list[2]);
                    off_bday.add(append_item);
                }else if(item.substring(0,15).equals("Office_friend: ")){
                    int l1 = item.length();
                    String[] String_list1 = item.substring(15,l1).split(",");
                    OfficeFriend append_item = new
OfficeFriend(item.substring(0,15), String_list1[0],  String_list1[1],
String_list1[2], String_list1[3]);
                    ofp_bday.add(append_item);
                }else if(item.substring(0,10).equals("Personal: ")){
                    int l2 = item.length();
                    String[] String_list2 = item.substring(10,l2).split(",");
                    Personal append_item = new Personal(item.substring(0,8),
String_list2[0],  String_list2[2],  String_list2[1], String_list2[3]);
                    pers_bday.add(append_item);
                }else{
                    System.out.println("error in intialize class logic");
                }
            }




            //ArrayLists to hold Mail Objects used for Serialization
            ArrayList<PersonalMail> mail_list1 = new ArrayList<PersonalMail>();
            ArrayList<GeneralMail> mail_list2 = new ArrayList<GeneralMail>();
            ArrayList<OfficeFriendMail> mail_list3 = new
ArrayList<OfficeFriendMail>();



            System.out.println("Please wait till we send the birthday mails");


            //sending out the birthday mails
            for(Personal item1 : pers_bday){
                if(item1.birthday.substring(4,10).equals(date_now.substring(4,
10))){
                    item1.SendBirthdayMail();
                    mail_list1.add(new PersonalMail(item1.email_address,"Birthday
Wish" , "hugs and love on your birthday. Navinda",
date_now,item1.nickname,item1.birthday));
                }
            }
```

```
            for(OfficeFriend item2 : ofp_bday){
                if(item2.birthday.substring(4,10).equals(date_now.substring(4,10)))
{
                    item2.SendBirthdayMail();
                    mail_list3.add(new
OfficeFriendMail(item2.email_address,"Birthday Wish" , "hugs and love on your
birthday. Navinda", date_now,item2.position,item2.birthday));
                }
            }



            //to take the deserialized objects
            ArrayList<GeneralMail> s11 = new ArrayList<GeneralMail>();
            ArrayList<PersonalMail> s12 = new ArrayList<PersonalMail>();
            ArrayList<OfficeFriendMail> s13 = new ArrayList<OfficeFriendMail>();

            //serializing birthday mails
            PersonalSerialize.serialize_start(mail_list1);
            OfficeFriendSerialize.serialize_start(mail_list3);
           //Serializing general mails
           General_serialize.serialize_start(mail_list2);
            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 recipient1
                    // store details in clientList.txt file
                    // Hint: use methods for reading and writing files
                    Scanner scanner2 = new Scanner(System.in);
                    System.out.println("write your new record \ninput format -
Official: nimal,nimal@gmail.com,ceo");
                    String entry = scanner2.nextLine();
                    FileHandle.WriteFile(entry);
                    scanner2.close();
                    break;
                  case 2:
                      // input format - email, subject, content
                      // code to send an email
                      Scanner scanner3 = new Scanner(System.in);
                      System.out.println("Input the Recepient Email");
                      String recepient = scanner3.nextLine();
                      System.out.println("Input the Subject");
                      String subject = scanner3.nextLine();
                      System.out.println("Input the Content");
                      String content = scanner3.nextLine();
                      SendEmailTLS.sendmail(recepient,subject,content);
                      mail_list2.add(new GeneralMail(recepient, subject, content,
date_now));
                      General_serialize.serialize_start(mail_list2);
                      scanner3.close();
                      break;
                  case 3:
                      // input format - yyyy/MM/dd (ex: 2018/09/17)
                      // code to print recipients who have birthdays on the given
date
                    Scanner scanner4 = new Scanner(System.in);
                    String day = scanner4.nextLine();
```

```
                    for(int i = 0; i < s;i++){
                        int size = birthday_list.get(i).length();
                        String compare = birthday_list.get(i).substring(size-10);
                        if(compare.equals(day)){
                            System.out.println(birthday_list.get(i));
                        }

                    }


                    scanner4.close();
                      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
                      try {
                        s11 = General_serialize.deserialize_start();
                      } catch (FileNotFoundException e) {
                        System.out.println("No General Mails Has been sent. Hence
not Deserialized");
                      }
                      //deserializing
                      s12 = PersonalSerialize.deserialize_start();
                      s13= OfficeFriendSerialize.deserialize_start();
                      System.out.println("Input the date you want to check\ninput
format - yyyy/MM/dd");
                      Scanner scanner5 = new Scanner(System.in);
                      String input_date = scanner5.nextLine();
                      for (GeneralMail item : s11){
                        if (item.date.equals(input_date)){
                            System.out.println(item.recepient+ "\n"+item.subject
+"\n"+ item.content);
                      }
                      }

                        for( PersonalMail item : s12){
                            if (item.date.equals(input_date)){
                                System.out.println(item.recepient+ "\
n"+item.subject +"\n"+ item.content);
                                System.out.println(item.nickname);
                                System.out.println(item.birthday);

                            }
                            }

                        for (OfficeFriendMail item : s13){
                            if (item.date.equals(input_date)){
                                System.out.println(item.recepient+ "\
n"+item.subject +"\n"+ item.content);
                                System.out.println(item.position);
                                System.out.println(item.birthday);

                            }
                            }

                      scanner5.close();
```

```
                    break;
                case 5:
                    // code to print the number of recipient objects in the
application
                    int x =InitializeClass.func();
                    System.out.println(x);
                    break;

            }



            scanner0.close();









        }
}

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









//SendEmailTLS code to send a mail
//fzwykuuekmjolcmz app password
// I have used methods in this class as static methods directly in the PSV main
above.

class SendEmailTLS {

    public static void sendmail(String recepient, String subject, String content) {

        final String username = "navindaoop@gmail.com";
        final String password = "fzwykuuekmjolcmz";
```

```
        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() {
                    protected PasswordAuthentication getPasswordAuthentication() {
                        return new PasswordAuthentication(username, password);
                    }
                });

        try {

            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress("navindaoop@gmail.com"));
            message.setRecipients(
                    Message.RecipientType.TO,
                    InternetAddress.parse(recepient)
            );
            message.setSubject(subject);
            message.setText(content);

            Transport.send(message);

            System.out.println("Done");

        } catch (MessagingException e) {
            e.printStackTrace();
        }
    }

}


//FileHandle class handle file writing and reading


class FileHandle {
    static ArrayList<String> file_objects = new ArrayList<String>();
    public static void WriteFile(String entry) {
        try {
            FileWriter writer = new FileWriter("clientList.txt",true);

            writer.write(entry);
            writer.write("\n");
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Error in FileHandle Class");

        }}
    public static ArrayList<String> ReadFile() {
        try {
            File myObj = new File("clientList.txt");
            Scanner myReader = new Scanner(myObj);
            while (myReader.hasNextLine()) {
```

```
              String data = myReader.nextLine();
              file_objects.add(data);
            }
            myReader.close();

          } catch (FileNotFoundException e) { //specific exception is handled here
            e.printStackTrace();
          }
          return file_objects;
    }
}



//RecepientSuperClass Superclass that gets inherited by 3 speicific recepient
sublasses
// I have used inheritance over here.
//good for code reusing
//also represents the template design pattern
abstract class RecepientSuperClass{
    protected String status;
    protected String name_of_recepient;
    protected String email_address;


}

//Birthday_interface which defines the birthday sending method for the recepient
subclasses
//this enables the official recepient to not define the birthday sending method and
still extend the Recepient SuperClass

interface Birthday_Interface {
    public void SendBirthdayMail();
}


//Personal Subclass
class Personal extends RecepientSuperClass implements Birthday_Interface {
    public String nickname;
    public String birthday;

    public Personal(String status,String name,String mail,String nickname, String
birthday){
        this.status = status;
        this.nickname = nickname;
        this.name_of_recepient = name;
        this.email_address = mail;
        this.birthday = birthday;

    }

    public void SendBirthdayMail(){
        SendEmailTLS.sendmail(this.email_address,"Birthday Wish","hugs and love on
your birthday. Navinda");
    }
}
```

```
//Official Subclass
class Official extends RecepientSuperClass{
    public String position;

    public Official(String status,String name,String mail,String position){
        this.position = position;
        this.name_of_recepient = name;
        this.status = status;
        this.email_address = mail;

    }


}


//office friend subclass
class OfficeFriend extends RecepientSuperClass implements Birthday_Interface {
    public String position;
    public String birthday;

    public OfficeFriend(String status,String name,String mail,String position,
String birthday){
        this.position = position;
        this.name_of_recepient = name;
        this.status = status;
        this.email_address = mail;
        this.birthday = birthday;

    }

    public void SendBirthdayMail(){
        SendEmailTLS.sendmail(this.email_address,"Birthday Wish","Wish you a Happy
Birthday. Navinda");
    }
}



//This class initializes all objects in the ClientList.txt when the application
starts
class InitializeClass {
    static ArrayList<RecepientSuperClass> int_objects = new
ArrayList<RecepientSuperClass>();
    static ArrayList<Personal> pers_objects = new ArrayList<Personal>();
    static ArrayList<OfficeFriend> ofp_objects = new ArrayList<OfficeFriend>();
    static ArrayList<Official> off_objects = new ArrayList<Official>();
    static ArrayList<String> pers_strings = new ArrayList<String>();
    public static int func(){
        ArrayList<String> st0 = FileHandle.ReadFile();
        for(String item : st0){
            String first = item.substring(0,10);
            if( first.equals("Official: ")){
                int l = item.length();
                String[] String_list = item.substring(9,l).split(",");
                Official append_item = new Official(item.substring(0,7),
String_list[0], String_list[1],  String_list[2]);
                int_objects.add(append_item);
                off_objects.add(append_item);
```

```
            }else if(item.substring(0,15).equals("Office_friend: ")){
                int l1 = item.length();
                String[] String_list1 = item.substring(14,l1).split(",");
                OfficeFriend append_item = new OfficeFriend(item.substring(0,12),
String_list1[0],  String_list1[1],  String_list1[2], String_list1[3]);
                int_objects.add(append_item);
                ofp_objects.add(append_item);
            }else if(item.substring(0,10).equals("Personal: ")){
                int l2 = item.length();
                String[] String_list2 = item.substring(10,l2).split(",");
                Personal append_item = new Personal(item.substring(0,8),
String_list2[0],  String_list2[2],  String_list2[1], String_list2[3]);
                int_objects.add(append_item);
                pers_objects.add(append_item);
                pers_strings.add(item);
            }else{
                System.out.println("error in intialize class logic");
            }

        }
        return int_objects.size();
    }
}

//Mail Object Super class made to be serializable
//provides a template for the extending sublclasses
// Hence the Template Design Pattern is used here
class MailClass implements Serializable {
    protected String recepient;
    protected String content;
    protected String subject;
    protected String date;

}

//general mail inheriting main MailClass
class GeneralMail extends MailClass{


    public GeneralMail(String recepient, String subject, String content, String
date){
        this.recepient = recepient;
        this.content = content;
        this.subject = subject;
        this.date = date;
    }
}


//office friend mail inheriting main MailClass
class OfficeFriendMail extends MailClass {

    public String position;
    public String birthday;


    public OfficeFriendMail(String recepient, String subject, String content,String
date,String position, String birthday){
        this.recepient = recepient;
```

```
        this.content = content;
        this.subject = subject;
        this.date = date;
        this.position = position;
        this.birthday = birthday;
    }
}

//personal mail exteding mail class
class PersonalMail extends MailClass{

    public String nickname;
    public String birthday;


    public PersonalMail(String recepient, String subject, String content,String
date,String nickname, String birthday){
        this.recepient = recepient;
        this.content = content;
        this.subject = subject;
        this.date = date;
        this.nickname = nickname;
        this.birthday = birthday;
    }
}


//Code to Serialize and Deserialize objects
class General_serialize{


    public static void serialize_start( ArrayList<GeneralMail> obj0) {
        try {
            FileOutputStream fileOut =new FileOutputStream("GeneralMail.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(obj0);
            out.close();
            fileOut.close();
            System.out.printf("Serialized data is saved in GeneralEmail.ser\n");
         } catch (IOException i) {
            i.printStackTrace();
         }
    }

    public static  ArrayList<GeneralMail> deserialize_start() throws
FileNotFoundException{
        ArrayList<GeneralMail> obj0 = new ArrayList<GeneralMail>();
        try {
            FileInputStream fileStream = new FileInputStream("GeneralMail.ser");
            ObjectInputStream os1 = new ObjectInputStream(fileStream);
            obj0 = (ArrayList) os1.readObject();
            os1.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return obj0;
    }
```

```
}

class OfficeFriendSerialize{


    public static void serialize_start( ArrayList<OfficeFriendMail> obj0) {
        try {
            FileOutputStream fileOut =new FileOutputStream("OfficeFriendMail.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(obj0);
            out.close();
            fileOut.close();
            System.out.printf("Serialized data is saved in OfficeEmails.ser\n");
         } catch (IOException i) {
            i.printStackTrace();
         }
    }

    public static ArrayList<OfficeFriendMail> deserialize_start() {
        ArrayList<OfficeFriendMail> obj0 = new ArrayList<OfficeFriendMail>();
        try {
            FileInputStream fileStream = new
FileInputStream("OfficeFriendMail.ser");
            ObjectInputStream os1 = new ObjectInputStream(fileStream);
            obj0 = (ArrayList) os1.readObject();
            os1.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return obj0;
    }}

class PersonalSerialize{


    public static void serialize_start( ArrayList<PersonalMail> obj0) {
        try {
            FileOutputStream fileOut =new FileOutputStream("PersonalMail.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(obj0);
            out.close();
            fileOut.close();
            System.out.printf("Serialized data is saved in PersonalMail.ser\n");
         } catch (IOException i) {
            i.printStackTrace();
         }
    }

    public static ArrayList<PersonalMail> deserialize_start() {
        ArrayList<PersonalMail> obj0 = new ArrayList<PersonalMail>();
        try {
            FileInputStream fileStream = new FileInputStream("PersonalMail.ser");
            ObjectInputStream os1 = new ObjectInputStream(fileStream);
            obj0 = (ArrayList) os1.readObject();
            os1.close();
        } catch (Exception e) {
            e.printStackTrace();
```

```
        }
        return obj0;
    }



}
```