```
//1. Address.java
public class Address
{
	private String streetAddress;
	private String city;
	private String state;
	private String zip;

	public Address()
	{
		streetAddress = "";
		city = "";
		state = "";
		zip = "";
	}

	public Address(String sAddress, String c, String s, String z)
	{
		streetAddress = sAddress;
		city = c;
		state = s;
		zip = z;
	}

	public void print()
	{
		System.out.println(streetAddress);
		System.out.println(city + ", " + state + " - " + zip);
	}

	public String toString()
	{
		return (streetAddress + "\n" + city + ", " + state + " - " +
zip);
	}

	public void setAddress(String sAddress, String c, String s, String z)
	{
		streetAddress = sAddress;
		city = c;
		state = s;
		zip = z;
	}

	public String getStreetAddress()
	{
		return streetAddress;
	}

	public String getCity()
	{
		return city;
	}

	public String getState()
	{
		return state;
```

```
	}

	public String getZip()
	{
		return zip;
	}

	public void copyAddress(Address otherAddress)
	{
		streetAddress = otherAddress.streetAddress;
		city = otherAddress.city;
		state = otherAddress.state;
		zip = otherAddress.zip;
	}
}

//2. ExtPerson.java
public class ExtPerson extends Person
{
	private Address address;
	private Date dob;
	private String phoneNumber;
	private String personStatus;

	public ExtPerson()
	{
		super("", "");
		address = new Address("", "", "", "");
		dob = new Date(1, 1, 1900);

		phoneNumber = "";
		personStatus = "";
	}

	public ExtPerson(String fName, String lName, int month, int day, int
year,
				String street, String c, String s, String z,
				String phone, String pStatus)
	{
		super(fName, lName);
		address = new Address(street, c, s, z);
		dob = new Date(month, day, year);

		phoneNumber = phone;
		personStatus = pStatus;
	}

	public void printAddress()
	{
		System.out.println(super.toString());
		System.out.println();
		System.out.println(address);
	}

	public void printInfo()
	{
		System.out.println(super.toString());
```

```
      System.out.println("Date of Birth: " + dob);

      System.out.println("Phone Number: " + phoneNumber);
      System.out.println("Person Type: " + personStatus);
      System.out.println(address);
}

public void setInfo(String fName, String lName,
                  int month, int day, int year,
                  String street, String c, String s,
                  String z, String phone, String pStatus)
{
      super.setName(fName,lName);
      dob.setDate(month, day, year);
      address.setAddress(street,c,s,z);
      phoneNumber = phone;
      personStatus = pStatus;
}

public void copyExtPerson(ExtPerson otherExtP)
{

       address.copyAddress(otherExtP.address);

     dob.setDate(otherExtP.dob.getMonth(),
             otherExtP.dob.getDay(),
             otherExtP.dob.getYear());

     phoneNumber = otherExtP.phoneNumber;
     personStatus = otherExtP.personStatus;
}

public boolean isLastName(String lName)
{
      return(super.getLastName().equals(lName));
}

public String getStatus()
{
      return personStatus;
}

public String getPhoneNumber()
{
      return phoneNumber;
}

public boolean isStatus(String status)
{
      return (status.equals(personStatus));
}

public boolean isDOB(int month, int day, int year)
{
      return(dob.getMonth() == month
             && dob.getDay() == day
```

```
                    && dob.getYear() == year);
      }

      public boolean isMonth(int month)
      {
            return(dob.getMonth() == month);
      }

      public int getMonth()
      {
            return dob.getMonth();
      }

      public int getDay()
      {
            return dob.getDay();
      }


      public int getYear()
      {
            return dob.getYear();
    }

    public String getStreetAddress()
      {
            return address.getStreetAddress();
      }

      public String getCity()
      {
            return address.getCity();
      }

      public String getState()
      {
            return address.getState();
      }

      public String getZip()
      {
            return address.getZip();
      }
}

//3. Person.java
public class Person
{
    private String firstName; //store the first name
    private String lastName;  //store the last name

          //Default constructor;
          //Initialize firstName and lastName to empty string
          //Postcondition: firstName = ""; lastName = "";
    public Person()
    {
          firstName = "";
```

```
        lastName = "";
    }

      //Constructor with parameters
      //Set firstName and lastName according to the parameters
      //Postcondition: firstName = first; lastName = last;
    public Person(String first, String last)
    {
           firstName = first;
         lastName = last;
    }

          //Method to output the first name and last name
          //in the form firstName lastName
    public String toString()
    {
            return (firstName + " " + lastName);
    }

            //Method to set firstName and lastName according to
            //the parameters
            //Postcondition: firstName = first; lastName = last;
    public void setName(String first, String last)
    {
         firstName = first;
         lastName = last;
    }

            //Method to return the firstName
          //Postcondition: the value of firstName is returned
    public String getFirstName()
    {
        return firstName;
    }

            //Method to return the lastName
          //Postcondition: the value of lastName is returned
    public String getLastName()
    {
        return lastName;
    }
}

//4. AddressBook.java
import java.io.*;

public class AddressBook
{
      private ExtPerson[] list;

      private int length;

      public AddressBook()
      {
            list = new ExtPerson[500];

            for(int i = 0; i < 500; i++)
```

```
            list[i] = null;


      length = 0;
}

public void print()
{
      for(int i = 0; i < length; i++)
            list[i].printInfo();
}

public void printNameInTheMonth(int month)
{
      for(int i = 0; i < length; i++)
      {
            if(list[i].isMonth(month))
                  System.out.println(list[i].getFirstName() + " " +
                                     list[i].getLastName());
      }
}

public void printInfoOf(String lName)
{
      int i = search(lName);

      if (i != -1)
            list[i].printInfo();
      else
            System.out.println(lName + " is not in address book.");
}

public void printNamesWithStatus(String status)
{
      for (int i = 0; i < length; i++)
            if (list[i].isStatus(status))
                  System.out.println(list[i].getFirstName() + " "
                                     + list[i].getLastName());
}

public void printAt(int i)
{
      if (i < length)
            list[i].printInfo();
      else
            System.out.println("No such person");

}

public void printNamesBetweenLastNames(String last1, String last2)
{
      String lName;

      for (int i = 0; i < length; i++)
      {
            lName = list[i].getLastName();
```

```
                if (last1.compareTo(lName) <= 0 && lName.compareTo(last2)
<= 0)
                      System.out.println(list[i].getFirstName() + " "
                                        + list[i].getLastName());
            }
      }

      public void insertAt(ExtPerson  eP, int i)
      {
            list[i] = null;
            list[i] = eP;
            if (i == length)
                  length++;

      }

      public void insertLast(ExtPerson eP)
      {
            list[length] = eP;
            length++;
      }

      public int search(String lName)
      {
            boolean found = false;
            int i;

            for (i = 0; i < length; i++)
                  if (list[i].isLastName(lName))
                  {
                        found = true;
                        break;
                  }

            if (found)
                  return i;
            else
                  return -1;
      }

      public void sort()
      {
            String str1;
            String str2;

            int i, j;

            ExtPerson temp = new ExtPerson();

            int minIndex;

            for (i = 0; i < length - 1; i++)
            {
                  minIndex = i;
                  str1 = list[minIndex].getLastName();

                  for (j = i + 1; j < length; j++)
```

```
                {
                        str2 = list[j].getLastName();
                        str1 = list[minIndex].getLastName();

                        if (str1.compareTo(str2) > 0)
                                minIndex = j;
                }

                temp.copyExtPerson(list[minIndex]);
                list[minIndex].copyExtPerson(list[i]);
                list[i].copyExtPerson(temp);
        }
    }

    public void saveData(PrintWriter outFile)
    {
        String first;
        String last;

        int month;
        int day;
        int year;

        String street;
        String city;
        String state;
        String zip;

        String phone;
        String pStatus;

        for (int i = 0; i < length; i++)
        {
                first = list[i].getFirstName();
                last = list[i].getLastName();
                month = list[i].getMonth();
                day = list[i].getDay();
                year = list[i].getYear();
                street = list[i].getStreetAddress();
                city = list[i].getCity();
                state = list[i].getState();
                zip = list[i].getZip();

                phone = list[i].getPhoneNumber();
                pStatus = list[i].getStatus();

                outFile.println(first + " " + last);
                outFile.println(month + " " + day + " " + year);
                outFile.println(street + " \n" + city + " \n" + state + "
\n" + zip);
                outFile.println(phone + " \n" + pStatus);
        }
    }

}

//5. Date.java
```

```
public class Date
{
   private int dMonth;        //variable to store the month
   private int dDay;          //variable to store the day
   private int dYear;         //variable to store the year

            //Default constructor
            //Data members dMonth, dDay, and dYear are set to
            //the default values
            //Postcondition: dMonth = 1; dDay = 1; dYear = 1900;
   public Date()
   {
        dMonth = 1;
        dDay = 1;
        dYear = 1900;
   }

            //Constructor to set the date
            //Data members dMonth, dDay, and dYear are set
            //according to the parameters
            //Postcondition: dMonth = month; dDay = day;
            //               dYear = year;
   public Date(int month, int day, int year)
   {
        dMonth = month;
        dDay = day;
        dYear = year;
   }

            //Method to set the date
            //Data members dMonth, dDay, and dYear are set
            //according to the parameters
            //Postcondition: dMonth = month; dDay = day;
            //                 dYear = year;
   public void setDate(int month, int day, int year)
   {
        dMonth = month;
        dDay = day;
        dYear = year;
   }

            //Method to return the month
            //Postcondition: The value of dMonth is returned
   public int getMonth()
   {
        return dMonth;
   }

            //Method to return the day
            //Postcondition: The value of dDay is returned
   public int getDay()
   {
        return dDay;
   }

            //Method to return the year
            //Postcondition: The value of dYear is returned
```

```
   public int getYear()
   {
        return dYear;
   }

            //Method to return the date in the form mm-dd-yyyy
   public String toString()
   {
            return (dMonth + "-" + dDay + "-" + dYear);
   }
}



//6. Ch10_PrExercise6.java

import java.io.*;
import java.util.*;

public class Ch10_PrExercise6
{
    static Scanner console = new Scanner(System.in);

    public static void main(String[] args) throws FileNotFoundException
    {
        AddressBook addressBook = new AddressBook();
        String str;
        String str1;
        String str2;
        int choice;
        int loc;
        int month;

        loadAddressBook(addressBook);

        addressBook.sort();

        showMenu();

        choice = console.nextInt();
        console.nextLine();

        while (choice != 9)
        {
            switch (choice)
            {
            case 1: System.out.print("Enter the last name of the person: ");
                    str = console.nextLine();
                    System.out.println();

                    loc = addressBook.search(str);

                    if (loc != -1)
                        System.out.println(str + " is in the address book");
                    else
```

```
                        System.out.println(str + " is not in the address
book");
                    break;
            case 2: System.out.print("Enter the last name of the person: ");
                    str = console.nextLine();
                    System.out.println();

                    loc = addressBook.search(str);

                    if (loc != -1)
                        addressBook.printAt(loc);
                    else
                        System.out.println(str + " is not in the address
book");
                    break;
            case 3: System.out.print("Enter the month number: ");
                    month = console.nextInt();
                    console.nextLine();
                    System.out.println();

                    addressBook.printNameInTheMonth(month);
                    break;
            case 4: System.out.print("Enter starting last name: ");
                    str1 = console.nextLine();
                    System.out.println();

                    System.out.print("Enter ending last name: ");
                    str2 = console.nextLine();
                    System.out.println();

                    addressBook.printNamesBetweenLastNames(str1, str2);
                    break;
            case 5: System.out.print("Enter person type Family, Friend,
Business: ");
                    str = console.nextLine();
                    System.out.println();

                    addressBook.printNamesWithStatus(str);
                    break;
            case 6: addressBook.print();
                    break;
            case 7: saveData(addressBook);
                    break;
            default: System.out.println("Invalid choice.");
            }

            showMenu();
            choice = choice = console.nextInt();
            console.nextLine();
        }


        char response;

        System.out.print("Save data Yes (Y/y) No(N/n)?: ");
        response = console.nextLine().charAt(0);
        System.out.println();
```

```
        if (response == 'y' || response == 'Y')
            saveData(addressBook);
    }

    public static void loadAddressBook(AddressBook adBook) throws

FileNotFoundException
    {
        Scanner inFile = new Scanner(new FileReader("Ch10_Ex6Data.txt"));

        String first;
        String last;

        int month;
        int day;
        int year;

        String street;
        String city;
        String state;
        String zip;

        String phone;
        String pStatus;

        ExtPerson temp;

        int i = 0;

        while (inFile.hasNext())
        {
            first = inFile.next();
            last = inFile.next();

            month = inFile.nextInt();
            day = inFile.nextInt();
            year = inFile.nextInt();
            inFile.nextLine();
            street = inFile.nextLine();
            city = inFile.nextLine();
            state = inFile.nextLine();
            zip = inFile.nextLine();
            phone = inFile.nextLine();
            pStatus = inFile.nextLine();

            temp = new ExtPerson();
            temp.setInfo(first, last, month, day, year,
                         street, city, state,
                         zip, phone, pStatus);

            adBook.insertAt(temp, i);
            i++;
        }
    }

    public static void saveData(AddressBook adBook) throws
```

```
                                                        FileNotFoundException
    {
        PrintWriter outfile;

        String filename;

        System.out.println("Enter file name: ");
        filename = console.nextLine();
        System.out.println();

        outfile = new PrintWriter(filename);

        adBook.saveData(outfile);
    }

    public static void showMenu()
    {
        System.out.println("Welcome to the address book program.");
        System.out.println("Choose among the following options:");
        System.out.println("1: To see if a person is in the address book");
        System.out.println("2: Print the information of a person");
        System.out.println("3: Print the names of person having birthday in
a particular month");
        System.out.println("4: Print the names of persons between two last
names");
        System.out.println("5: Print the names of persons having a
particular status");
        System.out.println("6: Print the address book");
        System.out.println("7: Save data");
        System.out.println("9: Terminate the program");
    }
}
```