```java
package model;

import java.text.DateFormat;

/**
 * Abstract superclass Person to store details common to Employee and Client.
 * @author Daria Vekic (Student ID: 586661)
 *
 */
public abstract class Person {

    //Instance fields
    protected String firstName;
    protected String lastName;
    protected Date dob;
    protected String addressLineOne;
    protected String addressLineTwo;
    protected String postcode;
    protected String phoneNum;
    protected DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");

    /**
     * Constructor method to set values of instance fields.
     * @param firstName - the first name of Employee/Client.
     * @param lastName - the last name of Employee/Client.
     * @param dob - the date of birth of Employee/Client.
     * @param addressLOne - the first line of address of Employee/Client.
     * @param addressLTwo - the second line of address of Employee/Client.
     * @param postcode - the postcode of Employee/Client.
     * @param phoneNum - the phone number of Employee/Client.
     * @throws ParseException - thrown if dob cannot be parsed.
     */
    public Person(String firstName, String lastName, String dob, String addressLOne, String addressLTwo,
            String postcode, String phoneNum) throws ParseException {
        this.firstName = firstName;
        this.lastName = lastName;
        this.dob = dateFormat.parse(dob);
        this.addressLineOne = addressLOne;
        this.addressLineTwo = addressLTwo;
        this.postcode = postcode.toUpperCase();
        this.phoneNum = phoneNum;
    } //end super constructor method

    /**
     * Method to access this Person's address line one.
     * @return addressLineOne
     */
    public String getAddressLineOne() {
        return addressLineOne;
    } //end method getAddressLineOne

    /**
     * Method to access this Person's address line two.
     * @return addressLineTwo
```

```
58      */
59     public String getAddressLineTwo() {
60         return addressLineTwo;
61     } //end method getAddressLineTwo
62 
63     /**
64      * Method to access this Person's date of birth.
65      * @return dob
66      */
67     public Date getDOB() {
68         return dob;
69     } //end method getDOB
70 
71     /**
72      * Method to access this Person's first name.
73      * @return firstName
74      */
75     public String getFirstName() {
76         return firstName;
77     } //end method getFirstName
78 
79     /**
80      * Method to access this Person's full name.
81      * @return the first name concatenated to last name
82      */
83     public String getFullName() {
84         return firstName + " " + lastName;
85     } //end method getFullName
86 
87     /**
88      * Method to access this Person's last name.
89      * @return lastName
90      */
91     public String getLastName() {
92         return lastName;
93     }//end method getFirstName
94 
95     /**
96      * Method to access this Person's phone number.
97      * @return phoneNum
98      */
99     public String getPhoneNum() {
100        return phoneNum;
101    } //end method getPhoneNum
102 
103    /**
104     * Method to access this Person's postcode.
105     * @return postcode
106     */
107    public String getPostcode() {
108        return postcode;
109    } //end method getPostcode
110 
111    /**
```

```
112      * Method to create a String comprised of this Person's details.
113      * @return this Person's details.
114      */
115     @Override
116     public String toString() {
117         return "First Name : " + firstName
118             + "\nLast Name : " + lastName
119             + "\nDob : " + dob
120             + "\nAddress : " + addressLineOne + ", " + addressLineTwo
121             + "\nPostcode : " + postcode
122             + "\nPhone Number : " + phoneNum;
123     } //end method toString
124 } //end class Person
```