```java
import java.text.*;
import javax.swing.JOptionPane;
import java.lang.Object;
import java.util.Random;
/*******************************************
 * MyPhone simulates the functionality of a smartphone.
 *
 * @author Shane Stacy
 * @version 10/8/2017
 *******************************************/
public class MyPhone
{

    /** integer for number of texts */
    private int numTexts;

    /** double for the amount of data consumed */
    private double dataConsumed;

    /** double for the remaining battery life */
    private double batteryLeft;

    /** string for the customer name */
    private String customerName;

    /** string for ten-digit phone number */
    private String phoneNumber;

    /** the audio usage per minute */
    private final double audioUsePerMinute = 65/60.0;

    /** the maximum minutes of audio usage for a full battery charge */
    private final double maxBatteryMinutes = 720.0;

    /*************************************
     * Constructor for phones
     * (make sure to use quotations in parameters!)
     * @param enterName the name of the customer
     * @param enterNumber the customer's phone number
     *************************************/
    public MyPhone (String enterName, String enterNumber) {

        //  initialize these variables for the object
        numTexts = 0;
        dataConsumed = 0.0;
        batteryLeft = 0.0;
        customerName = enterName;
        phoneNumber = enterNumber;

    }

    /*************************************
     * Get the number of texts
     *************************************/
    public int getNumTexts() {
        return numTexts;
    }

    /*************************************
     * Get the battery Life
     *************************************/
    public double getBatteryLife() {

        return batteryLeft;
    }

    /*************************************
     * Get the data usage in MB
     *************************************/
    public double getDataUsage() {

        return dataConsumed;
    }

    /*************************************
```

```
76       * Set the customer name
77       * @param n New name
78       ************************************/
79      public void setName (String n) {
80
81          customerName = n;
82
83          return;
84      }
85
86      /************************************
87       * Set the phone number
88       * @param n new number
89       ************************************/
90      public void setPhoneNumber (String n) {
91
92          phoneNumber = n;
93
94          if (phoneNumber.length() > 10) {
95              System.out.println("Error:  Phone Number cannot be greater than 10 digits. The Phone Number will now be
    9999999999.");
96              phoneNumber = "9999999999";
97              return;
98          }
99          return;
100     }
101
102     /************************************
103      * Method to discharge the battery
104      * @param mins minutes to charge the battery
105      ************************************/
106     public void chargeBattery (int mins) {
107
108         int charge = mins;
109         if (charge < 0) {
110             System.out.println("Error:  Negative input not valid for charging the battery.  Aborting method.");
111             return;
112         }
113         else if (batteryLeft < 1.0){
114             batteryLeft = batteryLeft + (charge * (100.0/12000.0));
115         }
116         JOptionPane displayBattery = new JOptionPane();
117         JOptionPane.showMessageDialog(displayBattery, "The battery is at " + (getBatteryLife() * 100.0) + "%.");
118         return;
119     }
120
121     /************************************
122      * Method to stream audio, uses data
123      * @param mins minutes of audio streaming
124      ************************************/
125     public void streamAudio (int mins) {
126
127         double possibleMins = batteryLeft * maxBatteryMinutes;
128         if (mins == 0) {
129             return;
130         }
131         if (mins < 0) {
132             System.out.println("Error:  Negative input not valid for streaming audio.  Aborting method.");
133             return;
134         }
135         if (mins <= possibleMins) {
136             batteryLeft = batteryLeft - (mins/maxBatteryMinutes);
137             dataConsumed = dataConsumed + (mins * audioUsePerMinute);
138         }
139         else {
140             dataConsumed = dataConsumed + (possibleMins * audioUsePerMinute);
141             batteryLeft = 0.0;
142             JOptionPane displayBatWarning = new JOptionPane();
143             JOptionPane.showMessageDialog(displayBatWarning, "The battery is empty.  Charge it.");
144         }
145         return;
146     }
147
148     /************************************
149      * Sending a text adds to text counter
```

```
150      * @param text text to send
151      ************************************/
152     public void sendText (String text) {
153 
154         String textDisplay = text;
155         JOptionPane textDialog = new JOptionPane();
156         JOptionPane.showInputDialog(textDialog, textDisplay);
157         numTexts = numTexts + 1;
158 
159         return;
160     }
161 
162     /**************************************
163      * Read the text
164      ************************************/
165     public void readText () {
166         Random rand = new Random();
167         int choice = rand.nextInt(5);
168         JOptionPane displayText = new JOptionPane();
169 
170         switch (choice) {
171             case 1:  JOptionPane.showMessageDialog(displayText, "Spicy memes.");
172             break;
173             case 2:  JOptionPane.showMessageDialog(displayText, "Pineapple pizza is the best.");
174             break;
175             case 3:  JOptionPane.showMessageDialog(displayText, "Gotta go fast.");
176             break;
177             case 4:  JOptionPane.showMessageDialog(displayText, "Remove the headphone jack.  Think Different.");
178             break;
179             case 5:  JOptionPane.showMessageDialog(displayText, "Have you played your Atari, today?");
180             break;
181         }
182 
183         return;
184     }
185 
186     /**************************************
187      * Print a phone statement and start a new month
188      ************************************/
189     public void printStatement() {  // print the phone statement
190 
191         NumberFormat fmt = NumberFormat.getCurrencyInstance();
192         System.out.println("Customer :            " + customerName);
193         System.out.println("Number:               " + fmtPhoneNumber());
194         System.out.println("Texts:                " + numTexts + " Texts");
195         System.out.println("Data usage:           " + dataConsumed + " MB");
196         System.out.println("");
197         System.out.println("2GB Plan:             " + fmt.format(50.00));
198         System.out.println("Additional data fee:  " + fmt.format(calcAdditionalDataFee()));
199         System.out.println("Universal Usage (3%): " + fmt.format(0.03 * 50.00));
200         System.out.println("Administrative Fee:   " + fmt.format(0.61));
201         System.out.println("Total Charges:        " + fmt.format(calcTotalFee()));
202         startNewMonth();
203 
204         return;
205     }
206 
207     /**************************************
208      * Set variables to zero for new month
209      ************************************/
210     private void startNewMonth() {
211         numTexts = 0;
212         dataConsumed = 0;
213         return;
214     }
215 
216     /**************************************
217      * Calculate the additional fee
218      ************************************/
219     private double calcAdditionalDataFee() {
220 
221         double fee = 0.0;
222         if (dataConsumed > 2000) {
223             int gigs = (((int)dataConsumed) / 1000) - 1;
224             fee = gigs * 15;
```

```
            return fee;
        }
        return fee;
    }

    /**************************************
     * Get the usage fee
     **************************************/
    private double calcUsageCharge() {
        double usageCharge;
        usageCharge = 0.03 * 50;
        return usageCharge;
    }

    /**************************************
     * Get the total fee
     **************************************/
    private double calcTotalFee() {
        double adminFee = 0.61;
        return  calcAdditionalDataFee() + calcUsageCharge() + 50.0 + adminFee;
    }

    /**************************************
     * Reformat the phone number to (###)###-####
     **************************************/
    private String fmtPhoneNumber() {
        String str = "(" + phoneNumber.substring(0,3) + ")" + phoneNumber.substring(3,6) + "-" +
phoneNumber.substring(6,10);
        return str;
    }

}
```