```java
import java.text.DecimalFormat;
import javax.swing.JOptionPane;
import java.util.Random;
/***************************************************************************
 * This program simulates a cell phone with data, battery, and texting.
 *
 * @author Ben Parsell
 * @version 10/1/15
 **************************************************************************/
public class MyPhone
{
    private int numTexts; // Total number of texts
    private double dataConsumed; // Data consumed in megabytes
    private double batteryLife; // Battery Life in a percentage
    private String customerName; // Customer name
    private String phoneNumber; // Customer phone number
    private boolean wifiOn; // Holds whether wifi is on/off
    private final double VIDEO_DATA_PER_MIN = 250 / 60.0; // Rate for video data
    private final double audio_data_per_min = 65 / 60.0; // Rate for audio data
    private final double audio_minutes = 720.0; // Total audio minutes on full charge
    private final double  video_minutes = 360.0; // Total video minutes on full charge

    /************************************************************
    Constructor for objects of class MyPhone
    This is the default constructor

    @param name customer name
    @param num customer phone number
    ************************************************************/
    public MyPhone (String name, String num) {
        customerName = name;
        phoneNumber = num;
        batteryLife = 0.0;
        dataConsumed = 0.0;
        if(phoneNumber.length() != 10){
            phoneNumber = "9999999999";
        }
    }


    /************************************************************
    Change the Customer Name.
    This is a mutator method
    @param n local variable for customer name
    ************************************************************/
    public void setName (String n) {
       customerName = n;
    }


    /***********************************************************
    Changes the customer's phone number.
    This is a mutator method

    @param n local variable for phone number
    ************************************************************/
    public void setPhoneNumber (String n) {
        phoneNumber = n;
    }


    /**********************************************************
    Formats the given phone number to standard phone number
    formatting.
    This is a mutator method.
    ***********************************************************/
    private String fmtPhoneNumber() {
        phoneNumber = "("  + phoneNumber.substring(0,3) + ")" + phoneNumb
er.substring(3,6) + "-" + phoneNumber.substring(6,9);
        return phoneNumber;
    }


    /**********************************************************
    Outputs the number of total texts.
    This is a mutator method

    @return number of total texts
    ***********************************************************/
    public int getNumTexts() {
        return numTexts;
    }


    /*********************************************************
    Outputs the battery life percentage.
    This is an accessor method.

    @return battery life percent
    ***********************************************************/
    public double getBatteryLife() {
```

```java
        return batteryLife;
    }


    /**********************************************************
    Outputs the total megabytes of data used
    This is an accessor method

    @return total data consumed in megabytes
    **********************************************************/
    public double getDataUsage() {
        return dataConsumed;
    }


    /*********************************************************
    Charges to battery — dependent on minutes from input
    This is a mutator method

    @param mins local variable for minutes
    *********************************************************/
    public void chargeBattery(int mins) {

        if((((batteryLife * 120) + mins) / 120) > 1.0) {
            batteryLife = 1.0;
        }else if(batteryLife > 0) {
            batteryLife = ((batteryLife * 120) + mins) / 120.0;
        }else if(mins <= 120) {
            batteryLife = mins / 120.0;
        } else if(mins > 120) {
            batteryLife = 1.0;
        }
        DecimalFormat fmt = new DecimalFormat("#.0");
        fmt.format(batteryLife);
        JOptionPane.showMessageDialog(null, "Battery Life: " + (fmt.format((batteryLife)*100)) + "%");
    }


    /*********************************************************
    Controls the wifi being on/off using boolean
    This is a mutator method

    *********************************************************/
    public void setWifi (boolean wifi) {
        wifiOn = wifi;
    }


    /*********************************************************
    Determines the amount of data consumed from streaming
    audio — dependent on minutes from parameter
    Also handles wifi

    @param mins local variable for minutes
    @return returns only for error catch (mins <0)
    *********************************************************/
    public void streamAudio(int mins) {
       if(mins < 0) {
           return;
        }

       if(wifiOn == true) {
           dataConsumed = dataConsumed + 0;
           batteryLife = batteryLife - (mins/audio_minutes);
           if(batteryLife <= 0) {
               batteryLife = 0;
            }
        }
       else if(wifiOn != true) {
           if((mins) / audio_minutes > 1 ) {
           dataConsumed = dataConsumed + 780;
           batteryLife = 0.0;

           } else if((mins) / audio_minutes > (batteryLife)) {
           dataConsumed = dataConsumed + ((audio_minutes*batteryLife) * audio_data_per_min);
           batteryLife = 0.0;
           JOptionPane.showMessageDialog(null, "Not enough battery for that many minutes. Phone will stream until battery is dead.");
           } else {
           batteryLife = batteryLife - (mins/audio_minutes);
           dataConsumed = dataConsumed + (mins*audio_data_per_min);
           }
       }
    }

    /*********************************************************
    Determines the amount of data consumed from streaming
    video — dependent on minutes from parameter
    Also handles wifi

    @param mins local variable for minutes
    @return returns only for error catch (mins <0)
    *********************************************************/
    public void streamVideo(int mins) {
        if(mins < 0) {
            return;
        }

        if(wifiOn) {
           dataConsumed = dataConsumed;
           batteryLife = batteryLife - (mins/video_minutes);
           if(batteryLife <= 0) {
               batteryLife = 0;
            }
        } else if(wifiOn != true) {
            if((mins) / video_minutes > 1 ) {
           dataConsumed = dataConsumed + 1500;
           batteryLife = 0.0;
           JOptionPane.showMessageDialog(null, "Not enough battery for th
at many minutes. Phone will stream until battery is dead.");
           } else if((mins) / video_minutes > (batteryLife)) {
           dataConsumed = dataConsumed + ((video_minutes*batteryLife) * V
IDEO_DATA_PER_MIN);
           batteryLife = 0.0;
           JOptionPane.showMessageDialog(null, "Not enough battery for th
at many minutes. Phone will stream until battery is dead.");
           } else {
           batteryLife = batteryLife - (mins/video_minutes);
           dataConsumed = dataConsumed + (mins*VIDEO_DATA_PER_MIN);
           }
        }
    }


    /*********************************************************
    Sends a text message and increments the total text count
    This is a mutator method

    @param text local variable for text message
    *********************************************************/
    public void sendText(String text) {
        if(batteryLife > 0) {
            numTexts += 1;
            JOptionPane.showMessageDialog(null, "Message sent!");
        }else if(batteryLife < 0) {
            JOptionPane.showMessageDialog(null, "Battery is dead.");
        }
    }


    /*********************************************************
    Reads a text message randomly selected from a switch
    statement. This uses the Random class.
    This is a mutator method.

    ***********************************************************/
    public void readText() {
        Random rand = new Random();
        int choice = rand.nextInt(5);
        if(batteryLife == 0.0) {
            JOptionPane.showMessageDialog(null, "You need to charge your phone to get texts");
        }else {
            switch (choice) {
                case 0:
                    JOptionPane.showMessageDialog(null, "What are you up to?");
                    break;

                case 1:
                    JOptionPane.showMessageDialog(null, "How's your day going?");
                    break;

                case 2:
                    JOptionPane.showMessageDialog(null, "Want to go grab pizza?");
                    break;

                case 3:
                    JOptionPane.showMessageDialog(null, "Buddy the elf, what's your favorite color?");
                    break;

               case 4:
                    JOptionPane.showMessageDialog(null, "What's your favorite movie?");
                    break;
            }
            numTexts += 1;
       }
    }


    /***********************************************************
    Prints out the final customer statement.
    MyPhoneTest uses this method to print final.
    This is a mutator method.

    ***********************************************************/
    public void printStatement() {
        DecimalFormat fmt = new DecimalFormat("#.0");
        System.out.println("MyPhone Monthly Statement");
        System.out.println("");
        System.out.println("Customer: \t\t" + customerName);
        System.out.println("Number: \t\t" + fmtPhoneNumber());
        System.out.println("Texts: \t\t\t" + numTexts);
        System.out.println("Data usage: \t\t" + fmt.format(dataConsumed/1
024) + " GB");
        System.out.println("");
        System.out.println("2GB Plan: \t\t" + "$50.00");
        System.out.println("Additional data fee: \t" + "$" + calcAddition
alDataFee());
        System.out.println("Universal Usage (3%): \t" + "$" + calcUsageCh
arge());
        System.out.println("Administrative Fee:\t$0.61");
        System.out.println("Total Charges: \t\t" + "$" + calcTotalFee());
    }


    /**********************************************************
    Resets number of texts and data consumed to start a new
    month. This is a private helper method.

    **********************************************************/
    private void startNewMonth() {
        dataConsumed = 0.0;
        numTexts = 0;
    }


    /**********************************************************
    Calculates any overuseage charges for data.
    This is a private helped method.

    @return extraCost returns final cost value.
    **********************************************************/
    private double calcAdditionalDataFee() {
      double extraCost = 0; // Final extra cost value
      double gbs; // Gigabytes
      gbs = Math.ceil(dataConsumed / 1024);
      if(dataConsumed > 2.0) {
          extraCost = Math.ceil(gbs - 2) * 15;
        }
      return extraCost;
    }


    /**********************************************************
    Calculates the 3% usage charge for the total bill cost.
    This is a private helper method.

    @return final value for usage charge cost.
    **********************************************************/
    private double calcUsageCharge() {
       return (50.0 + calcAdditionalDataFee())* .03 ;
    }


    /**********************************************************
    Calculates the total bill cost
    This is a private helper method

    @return final phone bill cost
    **********************************************************/
    private double calcTotalFee() {
       double fee = 50.0 + calcAdditionalDataFee();
       fee = fee + calcUsageCharge();
       fee = fee + 0.61;
       return fee;
    }
}
```