```java
 import java.util.Scanner;

class elecbill {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("--- Electricity Bill Generator ---");

        System.out.println("Enter Consumer Number:");
        String consumerNoInput = scanner.nextLine();
        if (consumerNoInput == null || consumerNoInput.equals("")) {
            System.out.println("Error: Consumer number cannot be empty.");
            scanner.close();
            return;
        }

        for (int i = 0; i < consumerNoInput.length(); i++) {
            if (!Character.isDigit(consumerNoInput.charAt(i))) {
                System.out.println("Error: Consumer number must be numeric.");
                scanner.close();
                return;
            }
        }
        int consumer_no = Integer.parseInt(consumerNoInput);

        System.out.println("Enter Consumer Name:");
        String consumer_name = scanner.nextLine();
        if (consumer_name == null) {
            System.out.println("Error: Consumer name cannot be empty.");
            scanner.close();
            return;
        }

        System.out.println("Enter Previous Month's Reading:");
        String prevInput = scanner.nextLine();
        if (prevInput == null || prevInput.equals("")) {
            System.out.println("Error: Previous reading cannot be empty.");
            scanner.close();
            return;
        }
        for (int i = 0; i < prevInput.length(); i++) {
            if (!Character.isDigit(prevInput.charAt(i))) {
                System.out.println("Error: Previous reading must be numeric.");
                scanner.close();
                return;
            }
        }
        int prev_reading = Integer.parseInt(prevInput);

        System.out.println("Enter Current Month's Reading:");
        String currInput = scanner.nextLine();
        if (currInput == null || currInput.equals("")) {
            System.out.println("Error: Current reading cannot be empty.");
            scanner.close();
            return;
        }
        for (int i = 0; i < currInput.length(); i++) {
            if (!Character.isDigit(currInput.charAt(i))) {
                System.out.println("Error: Current reading must be numeric.");
                scanner.close();
                return;
            }
        }
        int curr_month_reading = Integer.parseInt(currInput);

        if (prev_reading < 0 || curr_month_reading < 0) {
            System.out.println("Error: Readings cannot be negative.");
            scanner.close();
            return;
        }

        if (prev_reading > curr_month_reading) {
            System.out.println("Error: Previous reading cannot be greater than current 
reading.");
            scanner.close();
            return;
        }

        System.out.println("Enter connection type (domestic/commercial):");
        String connect_type = scanner.nextLine();
        if (connect_type == null || connect_type.equals("")) {
            System.out.println("Error: Connection type cannot be empty.");
            scanner.close();
            return;
        }

        int units = curr_month_reading - prev_reading;
        double billamt = 0.0;

        if (connect_type.equalsIgnoreCase("domestic")) {
            if (units <= 100) {
                billamt = units * 1.0;
            } else if (units <= 200) {
                billamt = 100 * 1.0 + (units - 100) * 2.50;
            } else if (units <= 500) {
                billamt = 100 * 1.0 + 100 * 2.50 + (units - 200) * 4.0;
            } else {
                billamt = 100 * 1.0 + 100 * 2.50 + 300 * 4.0 + (units - 500) * 6.0;
            }
        } else if (connect_type.equalsIgnoreCase("commercial")) {
            if (units <= 100) {
                billamt = units * 2.0;
            } else if (units <= 200) {
                billamt = 100 * 2.0 + (units - 100) * 4.50;
            } else if (units <= 500) {
                billamt = 100 * 2.0 + 100 * 4.50 + (units - 200) * 6.0;
            } else {
                billamt = 100 * 2.0 + 100 * 4.50 + 300 * 6.0 + (units - 500) * 7.0;
            }
        } else {
            System.out.println("Invalid Connection Type!");
            scanner.close();
            return;
        }

        System.out.println("\nELECTRICITY BILL");
        System.out.println("Consumer Number      : " + consumer_no);
        System.out.println("Consumer Name        : " + consumer_name);
        System.out.println("Connection Type      : " + connect_type);
        System.out.println("Previous Month Reading : " + prev_reading);
        System.out.println("Current Month Reading  : " + curr_month_reading);
        System.out.println("Units Consumed       : " + units);
        System.out.printf("Total Amount Payable : Rs. %.2f\n", billamt);

        scanner.close();
    }
}
```

```
--- Electricity Bill Generator ---
Enter Consumer Number:
321
Enter Consumer Name:
kumar
Enter Previous Month's Reading:
1234
Enter Current Month's Reading:
3545
Enter connection type (domestic/commercial):
domestic

ELECTRICITY BILL
Consumer Number      : 321
Consumer Name        : kumar
Connection Type      : domestic
Previous Month Reading : 1234
Current Month Reading  : 3545
Units Consumed       : 2311
Total Amount Payable : Rs. 12416.00
vangduk@vangduksubject:~/Desktop/vin$ java elecbill
--- Electricity Bill Generator ---
Enter Consumer Number:
4323
Enter Consumer Name:
kumaar
Enter Previous Month's Reading:
2654
Enter Current Month's Reading:
2225
Error: Previous reading cannot be greater than current reading.
vangduk@vangduksubject:~/Desktop/vin$ java elecbill
--- Electricity Bill Generator ---
Enter Consumer Number:
26437
Enter Consumer Name:
tyagi
Enter Previous Month's Reading:
3543
Enter Current Month's Reading:
3590
Enter connection type (domestic/commercial):
commercial

ELECTRICITY BILL
Consumer Number      : 26437
Consumer Name        : tyagi
Connection Type      : commercial
Previous Month Reading : 3543
Current Month Reading  : 3590
Units Consumed       : 47
Total Amount Payable : Rs. 94.00
vangduk@vangduksubject:~/Desktop/vin$ java elecbill
--- Electricity Bill Generator ---
Enter Consumer Number:
2241
Enter Consumer Name:
gyan
Enter Previous Month's Reading:
6342
Enter Current Month's Reading:
9989
Enter connection type (domestic/commercial):
commercial

ELECTRICITY BILL
```

```
Consumer Number      : 2241
Consumer Name        : gyan
Connection Type      : commercial
Previous Month Reading : 6342
Current Month Reading  : 9989
Units Consumed       : 3647
Total Amount Payable : Rs. 24479.00
vangduk@vangduksubject:~/Desktop/vin$ java elecbill
--- Electricity Bill Generator ---
Enter Consumer Number:
463
Enter Consumer Name:
ojas
Enter Previous Month's Reading:
1000
Enter Current Month's Reading:
6774
Enter connection type (domestic/commercial):
domestic

ELECTRICITY BILL
Consumer Number      : 463
Consumer Name        : ojas
Connection Type      : domestic
Previous Month Reading : 1000
Current Month Reading  : 6774
Units Consumed       : 5774
Total Amount Payable : Rs. 33194.00
```