

Java case study-1:

Group Name: Java Chips

Group Number: 3

Group Leader: KV Praseeda (2024000360)

Group Members:

2024000360 KV Praseeda

2024067820 Satyasiva Vasundhara

2024032550 Matta Lahari

2024007310

Project Name: Electricity Bill Calculator

Project Guide : DR.K. Naveen Kumar

Project Description:

- The electricity bill of an organization has to be calculated, according to the different slabs placed.
- For example, for the first 100 units, if Rs. 2 is taken, for the next 100, if Rs. 3 is taken, we must calculate the bill for the total units consumed.
- A simple calculation can be performed as:
- Let us assume 300 units are consumed by an entity.
This means that the total bill is calculated as:

Total Bill = (First 100 units*price slab 1) + (Next 100 units*price slab 2)
--

Approach to solving the problem:

- An easy way to solve this is by *dividing the slabs efficiently*
- Input: We take the total number of units consumed, as **t**.
- It will be taken as a floating point, in case fractional units are consumed.
- For this, we can mention the *number of slabs we want to add*. Assume it is **n**.
- Then, we use a *for loop*, to set the rate for each slab, for a set number of units
- Boundary condition:
If the number of units/number of slabs==1:
 We display the total power bill as $t*n$
If the number of units<0:
 We say it is invalid
- We will be using dynamic programming approach for this.

Explanation of execution:

- **SLABS:**

1. Quantity as a variable is declared(that is, the cap of units for each slab)
2. NUMBER OF SLABS TAKEN; n
3. FROM 0 TO n, USE SWITCH CASE, take quantity, and rate for each slab.
4. USE SWITCH CASE AGAIN:

CASE 1: FROM 0 TO SLAB 1:

1. Calculate the rate for the first slab

CASE 2: FROM (SLAB 1+1) TO SLAB 2:

1. Calculate rate of both first slab and second slab, and add both
2. If units still remain, we move to CASE 3

CASE 3: FROM (SLAB 2+1) TO N:

1. Calculate the bill till the highest rate, and add the prices from the previous slabs

DEFAULT CASE:

1. Display an error message

WHY SWITCH CASE WAS USED

- Allows to calculate the bill by case by case scenarios
- It is an example of modularity, where all problems are divided into sub-problems, thus making the calculation less complex.
- Allows for easy extension, if we consider that we want to increase the number of slabs/ change the rates

POTENTIAL DRAWBACKS+ SOLUTIONS FOR THE PREVIOUS APPROACH

- There are two situations related to solving this problem:
 1. If the administrator was running the system
 2. If the consumer wants to predict the electricity bill
- In case of the latter scenario, the bill might be easily tampered.
- So, an additional feature can be added.
- For now, let us use default slabs and rates. In case we want to change the rates, we can do the following:
 1. We can make it safer for the administrator to access, and modify the rates to values other than the default slabs. A conditional statement can be used, which requires them to enter a password. If it matches, the administrator can access and modify the rates. Otherwise, they cannot. Once the rates are modified, a sample bill is generated. After that, it asks the administrator the rate slabs must be changed again.
If yes, then the loops run again
If no, the modified rates are saved, then it again goes back to asking whether they are the admin or consumer.
 2. In case the user is a consumer, they can directly deposit the units consumed, and get the bill, according to the power consumed by them, and the rates set, or modified by the admin. They cannot modify the rate slabs.

PSEUDOCODE

START

```
slabCap[] = {100, 200, 500}    // default caps
```

```
rate[] = {5.0, 7.0, 10.0}    // default rates
```

```
ADMIN_PASSWORD = "admin123"
```

```
while(true) {
```

```
    print("1. Admin")
```

```
    print("2. Consumer")
```

```
    print("3. Exit")
```

```
    choice = input()
```

```
    switch(choice) {
```

```
        case 1: // Admin
```

```
            print("Enter password:")
```

```
            password = input()
```

```
            if(pass == ADMIN_PASSWORD) {
```

```
                print("Do you want to modify slabs? (Yes/No)")
```

```
                answer = input()
```

```
                if(ans == "Yes") {
```

```
                    for(i = 0; i < slabCap.length; i++) {
```

```
                        print("Enter new cap for slab " + (i+1))
```

```
                        slabCap[i] = input()
```

```
                        print("Enter new rate for slab " + (i+1))
```

```
                        rate[i] = input()
```

```
                    }
```

```
                    print("Slabs updated successfully")
```

```
                }
```

```
            else {
```

```
                print("Using existing slabs")
```

```
            }
```

```
        }
```

```

else {
    print("Incorrect password")
}

break

case 2: // Consumer

    print("Enter total units consumed")

    units = input()

    // calculate bill

    bill = 0

    caseNum = 0

    for(i = 0; i < slabCap.length; i++) {
        if(units <= slabCap[i]) {
            caseNum = i+1
            break
        }
    }

    if(caseNum == 0) {
        caseNum = slabCap.length
    }

    switch(caseNum) {
        case 1:
            bill = units * rate[0]

            break

        default:
            bill = (slabCap[0] * rate[0])

            for(i = 1; i < caseNum; i++) {
                if(i == caseNum-1) {
                    bill = bill + (units - slabCap[i-1]) * rate[i]
                }
            }

            else {

```

```
        bill = bill + (slabCap[i] - slabCap[i-1]) * rate[i]
    }
}

print("Total bill = " + bill)
break

case 3:
    print("Exit program")
    STOP
default:
    print("Invalid choice")
}
}
```

CODE:

```
import java.util.Scanner;
class ElectricityBillSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Default slabs (used unless admin modifies them)
        int[] slabCap = {100, 200, 500}; // unit caps
        double[] rate = {5.0, 7.0, 10.0}; // rates for each slab
        final String ADMIN_PASSWORD = "admin123"; // simple password
        boolean exit = false;
        while (!exit) {
            System.out.println("\n--- ELECTRICITY BILLING SYSTEM ---");
            System.out.println("1. Admin");
            System.out.println("2. Consumer");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();
            switch (choice) {
                case 1: // Admin
                    System.out.print("Enter Admin Password: ");
                    String pass = sc.next();
                    if (pass.equals(ADMIN_PASSWORD)) {
                        System.out.println("Login Successful! Current slabs are:");
                        for (int i = 0; i < slabCap.length; i++) {
                            System.out.println("Slab " + (i + 1) + ": up to " + slabCap[i] + " units @ Rs " + rate[i] +
                                "/unit");
                        }
                        System.out.print("\nDo you want to modify slabs? (Yes/No): ");
                        String modify = sc.next();
                        if (modify.equalsIgnoreCase("Yes")) {
                            for (int i = 0; i < slabCap.length; i++) {
                                System.out.print("Enter new cap for slab " + (i + 1) + ": ");
                                slabCap[i] = sc.nextInt();
                                System.out.print("Enter new rate for slab " + (i + 1) + ": ");
                                rate[i] = sc.nextDouble();
                            }
                            System.out.println("Slabs updated successfully!");

                            // After updating, take units and calculate bill
                            System.out.print("\nEnter total units consumed: ");
                            int units = sc.nextInt();
                        }
                    }
                }
            }
        }
    }
}
```

```

        double bill = calculateBill(units, slabCap, rate);
        System.out.println("Your total bill = Rs " + bill);
    } else {
        System.out.println("No changes made. Using existing slabs.");
    }

    // Directly proceed to bill calculation
    System.out.print("\nEnter total units consumed: ");
    int units = sc.nextInt();
    double bill = calculateBill(units, slabCap, rate);
    System.out.println("Your total bill = Rs " + bill);
}
} else {
    System.out.println("Incorrect password! Access denied.");
}
break;
case 2: // Consumer
    System.out.println("Using current slab rates (default if admin not logged in):");
    for (int i = 0; i < slabCap.length; i++) {
        System.out.println("Slab " + (i + 1) + ": up to " + slabCap[i] + " units @ Rs " + rate[i] +
"/unit");
    }
    System.out.print("\nEnter total units consumed: ");
    int units = sc.nextInt();

    double bill = calculateBill(units, slabCap, rate);
    System.out.println("Your total bill = Rs " + bill);
    break;
case 3: // Exit
    exit = true;
    System.out.println("Exiting... Thank you!");
    break;
default:
    System.out.println("Invalid choice. Please try again.");
}
}
}
sc.close();
}

```

```

// Function to calculate bill using switch-case logic
public static double calculateBill(int units, int[] slabCap, double[] rate) {
    double bill = 0;
    int caseNum = 0;

    // Find which case to apply
    for (int i = 0; i < slabCap.length; i++) {
        if (units <= slabCap[i]) {
            caseNum = i + 1;

```

```

        break;
    }
}
if (caseNum == 0) {
    caseNum = slabCap.length; // last case if units exceed all slabs
}

// Switch-case calculation
switch (caseNum) {
    case 1:
        bill = units * rate[0];
        break;

    default:
        bill = (slabCap[0] * rate[0]); // first slab always fully used
        for (int i = 1; i < caseNum; i++) {
            if (i == caseNum - 1) {
                bill += (units - slabCap[i - 1]) * rate[i];
            } else {
                bill += (slabCap[i] - slabCap[i - 1]) * rate[i];
            }
        }
    }
}
return bill;
}

```

OUTPUT:

Case 1: Admin access + slabs changed

```

--- ELECTRICITY BILLING SYSTEM ---
1. Admin
2. Consumer
3. Exit
Enter your choice: 1
Enter Admin Password: admin123
Login Successful! Current slabs are:
Slab 1: up to 100 units @ Rs 5.0/unit
Slab 2: up to 200 units @ Rs 7.0/unit
Slab 3: up to 500 units @ Rs 10.0/unit

Do you want to modify slabs? (Yes/No): yes
Enter new cap for slab 1: 15
Enter new rate for slab 1: 15
Enter new cap for slab 2: 28
Enter new rate for slab 2: 30
Enter new cap for slab 3: 88
Enter new rate for slab 3: 44
Slabs updated successfully!

Enter total units consumed: 7895
Your total bill = Rs 346763.0

```


Case 2: WHEN RATES MODIFIED, THEY STAY FOR THE ITERATION

```
--- ELECTRICITY BILLING SYSTEM ---
1. Admin
2. Consumer
3. Exit
Enter your choice: 2
Using current slab rates (default if admin not logged in):
Slab 1: up to 15 units @ Rs 15.0/unit
Slab 2: up to 28 units @ Rs 30.0/unit
Slab 3: up to 88 units @ Rs 44.0/unit

Enter total units consumed: 20
Your total bill = Rs 375.0
```

CASE 3: CONSUMER+DEFAULT RATES:

```
--- ELECTRICITY BILLING SYSTEM ---
1. Admin
2. Consumer
3. Exit
Enter your choice: 2
Using current slab rates (default if admin not logged in):
Slab 1: up to 100 units @ Rs 5.0/unit
Slab 2: up to 200 units @ Rs 7.0/unit
Slab 3: up to 500 units @ Rs 10.0/unit

Enter total units consumed: 45
Your total bill = Rs 225.0
```

CASE 4: ADMIN+DEFAULT RATES

```
--- ELECTRICITY BILLING SYSTEM ---  
1. Admin  
2. Consumer  
3. Exit  
Enter your choice: 1  
Enter Admin Password: admin123  
Login Successful! Current slabs are:  
Slab 1: up to 100 units @ Rs 5.0/unit  
Slab 2: up to 200 units @ Rs 7.0/unit  
Slab 3: up to 500 units @ Rs 10.0/unit  
  
Do you want to modify slabs? (Yes/No): no  
No changes made. Using existing slabs.  
  
Enter total units consumed: 66  
Your total bill = Rs 330.0
```

CASE 5: EXIT CASE

```
--- ELECTRICITY BILLING SYSTEM ---  
1. Admin  
2. Consumer  
3. Exit  
Enter your choice: 3  
Exiting... Thank you!
```

FUTURE SCOPE:

- We can manage a database of slab rates
- We can add multiple users, across different levels
- The interface can be improved
- This code helps the consumer to predict their electricity bills in real time, so it can help in judicious consumption of energy
- It also allows transparency, and can be implemented at a larger scale, by collaborating with the government.