School of Computer and Information Sciences

Software engineering Lab

I.M. Tech VI – Jan – May 2026

Task 1: Problem statement: Electricity Bill system

Design and implement to generate the electricity bill that performs the following

1. Input
    a. Accept the required details
    b. For each consumer, input to be taken at the time of execution
    c. Consumer / Service number is unique
    d. Names must be alphabets only, no special characters or numbers should be accepted
    e. Phone number length should be exactly 10 characters.
2. Validation
    a. Reject the duplicate consumer / service numbers
    b. Reject names with numbers and special characters
    c. Reject phone number if it is less than or greater than 10 characters
3. Computation
    a. Calculate the total number of units that are consumed and bill accordingly. Rate per unit 1. First 50 units – 1.5, 2. Second 50 units – 2.5, 3. Third fifty units – 3.5, 4. Later onwards 4.5.
    b. If the number of units consumed are 0, then minimum charge 25/- has to be levied.
    c. Fine – 150/-
4. Output
    a. Display the bill with consumer name, service number, date, number of units that are consumed, previous bill if any pending, due date to pay without fine, after due date with fine.
5. Additional requirements
    a. Implement modular programming (use functions for inputs, validations, computation and output).
    b. Ensure proper error handling – re prompt if invalid data is entered.

Task 2: Electricity Bill system

Modify the Electricity Bill system developed and generate another version of the same on git incorporating the following.

1. Modularization
    a. Make the user defined header files
2. Quality characters
    a. Usability – easy to use
    b. Efficiency – make optimal use of resources including memory, utilization, processing time, etc
    c. Reusability – reuse the modules that are developed across the applications
    d. Interoperability – try to invoke the functionalities of the applications from other platforms.
3. Documentation
    a. Write modules specifications – module name, input, preconditions, logic, output for each module
    b. Logic can be represented using an algorithm, flowchart or pseudo code
    c. Use draw.io to draw figures if any
4. Test plan
    a. Write test plan for each of the function you write in the application
    b. Test plan includes test id, functionality, input, expected output, actual output and test report.