

**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110  
(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**Date: 27.07.2024**

## **Mini Project**

### **Objective:**

1. To develop a real world application using Java, as a team project.
2. To use various object oriented concepts learnt in the course
3. To prepare a detailed report having the following components:
  - Problem statement
  - Motivation for the problem
  - Scope and Limitations
  - Design of the solution (class diagram)
  - Modules split-up
  - Implementation
  - Output screenshots
  - Object oriented features used
  - Inference and future extension
4. To present and demonstrate the project for evaluation

### **Best Practices:**

- Class Diagram usage
- Naming convention – for file names, variables
- Comment usage at proper places
- Prompt messages during reading input and displaying output
- Incremental program development
- Modularity
- All possible test cases in output

**Question:**

Please follow the instructions given below to develop a Java application that automates a real life system.

1. Describe the problem statement in detail specifying the inputs, output, scope and limitations. (CO3,K4)
2. Using various object oriented concepts learnt in the course, design the solution, and represent it using class diagrams. (CO2,K3)
3. Select the suitable design alternative. (CO3,K5)
4. Use various object oriented concepts learnt in the course. (CO2,K3)
5. Implement the design solution using Java. (CO3,K4)
6. Integrate and evaluate the project for different test cases. (CO3 K5)
7. Your project should have the following: (CO3,K4)
  - a. An interactive text-based interface to accept inputs and display outputs
  - b. A file-based storage for storing the data
  - c. A Java package that has distinct Java files for each functionality
  - d. A well-documented codebase
8. Prepare a detailed report with the following components: (CO3,K4)
  - a. Problem statement
  - b. Motivation for the problem
  - c. Scope and Limitations
  - d. Design of the solution (class diagram)
  - e. Modules split-up
  - f. Implementation specifics
  - g. Output screenshots
  - h. Object oriented features used
  - i. Inference and future extension
9. Present and demonstrate the project for evaluation with sufficient test cases that cover all use cases. (CO3,K4)
  - a. Include best practices, class diagrams for design, modularity, recommended naming conventions, well-documented codebase, prompt messages input and output, and incremental mode of program development.

### **Sample Problems for the Mini Project:**

You may use the following sample problems to get an idea about the application to be developed. You should get the problem statement approved by the course faculty before proceeding. We encourage you to consider diverse applications before finalizing your problem statement. You are advised to adopt a problem that can possibly be converted into a full product later.

1. **Online examination:** The system should have the following functionalities:
  - a. Configuration of a question repository of MCQ questions with options and correct answers, by an admin.
  - b. Attempting an exam by the students where the questions are populated randomly from the question repository.
  - c. Provision to display the rank list based on the scores in the exam.
2. **Train seat booking:** The system should have the following functionalities:
  - a. Configuration of available berths (coach number and seat numbers in each coach) and coach types (AC / sleeper) for at least five source-destination stations in a train, by an admin.
  - b. Booking tickets by passengers by giving coach type, source and destination.
  - c. Display the final chart of seat allocation.
3. **Online voting:** The system should have the following functionalities:
  - a. Configuration of the names of candidates, their constituency and their election symbol (this can be a simple text), by an admin.
  - b. Configuration of the names of voters, and their constituency, by an admin.
  - c. Voting by voters from among the list of symbols
  - d. Display the final list of candidates and the votes they received in decreasing order of votes received
4. **Grocery Bill Generation:** The system should have the following functionalities:
  - a. Configuration of name, category and prices per unit weight for each product, by an admin.
  - b. Adding custom quantities of multiple items to the basket, by the consumer.
  - c. Display the final bill of products for each order once the order is confirmed, for the consumer.
  - d. Display a total number of items sold under each category, for the admin.