



Sheryians Coding  
School

# Live Cohort

## ASSIGNMENT 1



**Q1. Define a JavaScript class named CabService that will be used to calculate the bill for a city cab ride. List and declare the following properties (data members) inside the class:**

- **car\_type (String): To store the type of car ("AC" or "NON AC")**
- **km (Number): To store the kilometers travelled**
- **bill (Number): To store the total bill**

**Q2. Write the constructor method for the CabService class that initializes:**

- **car\_type to an empty string**
- **km and bill to 0.0**

**Q3. Define a method accept(carType, km) inside the class that**

- **Takes carType and km as parameters**
- **Assigns them to the respective class properties**

**Q4. Define a method `calculate()` that calculates the total bill amount using the following logic:**

- For AC Cars:
  - Up to 5 KM → ₹150
  - Beyond 5 KM → ₹150 + ₹10 for every KM beyond 5
- For NON AC Cars:
  - Up to 5 KM → ₹120
  - Beyond 5 KM → ₹120 + ₹8 for every KM beyond 5

**Q5. Define a method `display()` that prints the details in the following format:**

CAR TYPE: <car\_type>

KILOMETER TRAVELED: <km>

TOTAL BILL: ₹<bill>

**Q6. Create an object named `cab1` of the class `CabService`.**

**Q7. Call the `accept()` method with user-defined input (e.g., "AC", 8).**

**Q8. Call the `calculate()` method to compute the total fare based on input.**

**Q9. Call the display() method to print the final bill.**





Sheryians Coding  
School

# Live Cohort

## ASSIGNMENT 2



# Assignment: JavaScript-Based Railway Ticket Booking System

## Questions:

### 1. Class Definition

- Define a class named RailwayTic

### 2. Data Members

- Inside the class, declare the following properties:
  - name: to store the name of the customer (string)
  - coach: to store the type of coach (string)
  - mob\_no: to store the mobile number (number)
  - amt: to store the base ticket amount (number)
  - total\_amt: to store the total amount after coach-based fare update (number)

### 3. Method: accept()

- Define a method accept(name, coach, mob\_no, amt) that assigns the given values to the respective data members of the class.

### 4. Method: update()

- Define a method update() that modifies the total\_amt based on the value of coach. Use the following fare chart:

## **5. Method: display()**

- Define a method **display()** that prints the ticket details in the following format:

**NAME: <name>**

**COACH: <coach>**

**MOBILE NUMBER: <mob\_no>**

**TOTAL AMOUNT: ₹<total\_amt>**

## **6. Create an object of the RailwayTicket class.**

## **7. User Interaction**

- Use the object to:

- Call the **accept()** method with sample input values.
- Call the **update()** method to calculate the total fare.
- Call the **display()** method to print the ticket details.