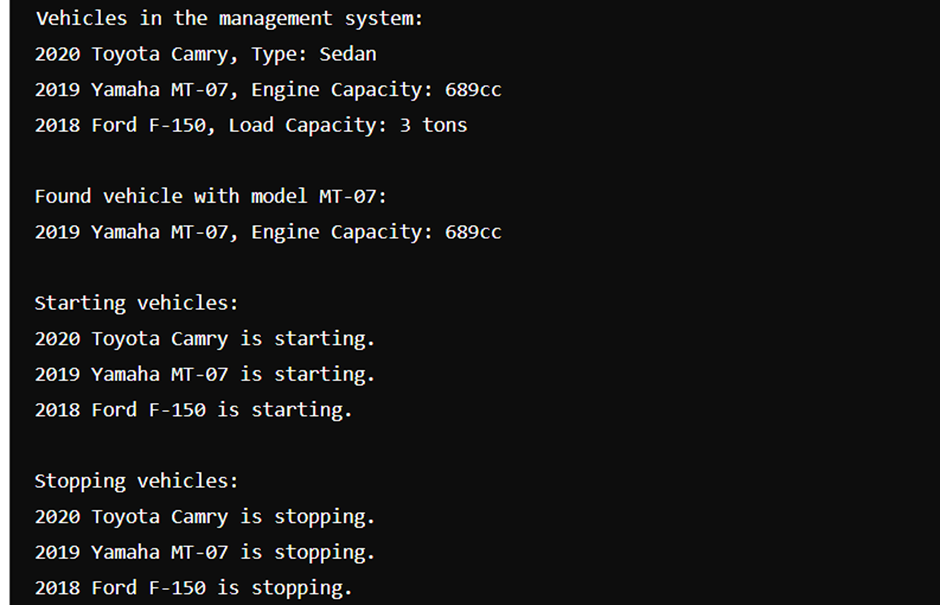
**Classroom Activity: Write a program to develop a vehicle management system using different types of vehicle having common behaviour**

**Code:**

|  |
| --- |
| // Base class for vehicles  class Vehicle {  constructor(brand, model, year) {  this.brand = brand;  this.model = model;  this.year = year;  }    getDetails() {  return `${this.year} ${this.brand} ${this.model}`;  }    start() {  return `${this.getDetails()} is starting.`;  }    stop() {  return `${this.getDetails()} is stopping.`;  }  }    // Derived class for Cars  class Car extends Vehicle {  constructor(brand, model, year, type) {  super(brand, model, year);  this.type = type;  }    getDetails() {  return `${super.getDetails()}, Type: ${this.type}`;  }  }    // Derived class for Motorcycles  class Motorcycle extends Vehicle {  constructor(brand, model, year, engineCapacity) {  super(brand, model, year);  this.engineCapacity = engineCapacity;  }    getDetails() {  return `${super.getDetails()}, Engine Capacity: ${this.engineCapacity}cc`;  }  }    // Derived class for Trucks  class Truck extends Vehicle {  constructor(brand, model, year, loadCapacity) {  super(brand, model, year);  this.loadCapacity = loadCapacity;  }    getDetails() {  return `${super.getDetails()}, Load Capacity: ${this.loadCapacity} tons`;  }  }    // Vehicle management system  class VehicleManagementSystem {  constructor() {  this.vehicles = [];  }    addVehicle(vehicle) {  this.vehicles.push(vehicle);  }    listVehicles() {  return this.vehicles.map(vehicle => vehicle.getDetails()).join('\n');  }    findVehicle(model) {  return this.vehicles.find(vehicle => vehicle.model === model);  }  }    // Create a new vehicle management system  const vms = new VehicleManagementSystem();    // Add vehicles to the system  const car1 = new Car('Toyota', 'Camry', 2020, 'Sedan');  const motorcycle1 = new Motorcycle('Yamaha', 'MT-07', 2019, 689);  const truck1 = new Truck('Ford', 'F-150', 2018, 3);    vms.addVehicle(car1);  vms.addVehicle(motorcycle1);  vms.addVehicle(truck1);    // List all vehicles in the system  console.log('Vehicles in the management system:');  console.log(vms.listVehicles());    // Find a vehicle by model  const modelToFind = 'MT-07';  const foundVehicle = vms.findVehicle(modelToFind);  console.log(`\nFound vehicle with model ${modelToFind}:`);  console.log(foundVehicle ? foundVehicle.getDetails() : 'Vehicle not found');    // Start and stop vehicles  console.log(`\nStarting vehicles:`);  console.log(car1.start());  console.log(motorcycle1.start());  console.log(truck1.start());    console.log(`\nStopping vehicles:`);  console.log(car1.stop());  console.log(motorcycle1.stop());  console.log(truck1.stop()); |

**Output:**



**Execution-Steps:**

* Create a JavaScript file with any name
* Write the code
* Run the code in the terminal using F5 or right click and go to inspect and you can get the output in the console as well.