MINI Project Summary
On
Object Oriented Programming

Topic/Title: **"Automobile Service Billing System"**

- **Introduction** :

1) "**Vehicle Servicing Billing System**" is designed to streamline the process of managing service orders and generating bills for vehicle maintenance.
2) It offers users the convenience of selecting from a range of services such as oil changes, tire rotations, ad car washes, with corresponding prices displayed for each service.
3) Utilizing object-oriented principles, the system employs hierarchical inheritance to categorize services under the abstract 'Service' class, allowing for easy extension to accommodate additional service types.
4) Abstraction is employed to encapsulate service details and display logic, promoting code modularity and flexibility.
5) The system also incorporates file handling functionality, enabling the storage of service order details in a text file for record-keeping purposes.
6) Through its intuitive user interface and robust backend functionality, the Vehicle Servicing Billing System provides an efficient solution for automotive service centers to manage customer orders and generate accurate bills.

- **Home Screen Snap**:

```
C:\Users\SANSKRITI>cd C:\Users\SANSKRITI\OneDrive\Desktop\Java Mini Project

C:\Users\SANSKRITI\OneDrive\Desktop\Java Mini Project>javac VehicleServicingBillingSystem.java

C:\Users\SANSKRITI\OneDrive\Desktop\Java Mini Project>java VehicleServicingBillingSystem
Enter owner's name: Sanskriti Kuwar
Enter vehicle plate: 789465
Choose a service by entering its corresponding number:
1. Regular Oil Change - Rs.500.0
2. Synthetic Oil Change - Rs.800.0
3. Standard Tire Rotation - Rs.1100.0
4. Balanced Tire Rotation - Rs.900.0
5. Exterior Wash - Rs.550.0
6. Interior and Exterior Wash - Rs.1200.0
Enter '0' to finish selection.
Enter the index of the service:
```

- **Conclusion/Learning**:

In conclusion, through the implementation of hierarchical inheritance, abstraction, and encapsulation, the codebase achieves modularity, extensibility, and maintainability and developed

"**Vehicle Servicing Billing System**". The hierarchical class structure allows for the categorization of services, facilitating the addition of new service types in the future. Abstraction helps in hiding implementation details, promoting code reuse and simplifying system maintenance. Additionally, encapsulation ensures data integrity by encapsulating service details within their respective classes. Furthermore, the inclusion of file handling functionality enables the system to persist service order details for record-keeping purposes.

Overall, developing this project provides valuable learning opportunities in designing and implementing object-oriented systems, handling user input, performing file operations, and managing service orders effectively. This project underscores the importance of leveraging object-oriented principles to build robust and scalable software solutions.

Complete Code:

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

abstract class Service {
    protected String name;
    protected double price;

    public Service(String name, double price) {
        this.name = name;
        this.price = price;
    }

    public abstract void display();
}

class OilChange extends Service {
    public OilChange(String name, double price) {
        super(name, price);
    }

    @Override
    public void display() {
        System.out.println("Oil Change Service: " + name + " - Rs." + price);
    }
}

class TireRotation extends Service {
    public TireRotation(String name, double price) {
        super(name, price);
    }
```

```
    @Override
    public void display() {
        System.out.println("Tire Rotation Service: " + name + " - Rs." +
price);
    }
}

class CarWash extends Service {
    public CarWash(String name, double price) {
        super(name, price);
    }

    @Override
    public void display() {
        System.out.println("Car Wash Service: " + name + " - Rs." + price);
    }
}

class VehicleServiceOrder {
    private List<Service> services;
    private String ownerName;
    private String vehiclePlate;

    public VehicleServiceOrder(String ownerName, String vehiclePlate) {
        this.ownerName = ownerName;
        this.vehiclePlate = vehiclePlate;
        services = new ArrayList<>();
    }

    public void addService(Service service) {
        services.add(service);
    }

    public void display() {
        System.out.println("Service Order for " + ownerName + "'s Vehicle
(Plate: " + vehiclePlate + "):");
        for (Service : services) {
            service.display();
        }
    }

    public double getTotalPrice() {
        double totalPrice = 0.0;
        for (Service : services) {
            totalPrice += service.price;
        }
        return totalPrice;
```

```
    }

    public void saveToFile(String filename) {
        try (FileWriter writer = new FileWriter(filename, true)) {
            writer.write("Owner Name: " + ownerName + "\n");
            writer.write("Vehicle Plate: " + vehiclePlate + "\n");
            writer.write("Services:\n");
            for (Service : services) {
                writer.write(service.name + " - Rs." + service.price + "\n");
            }
            writer.write("Total Price: Rs." + getTotalPrice() + "\n\n");
            System.out.println("Service order details appended to " +
filename);
        } catch (IOException e) {
            System.out.println("Failed to save service order details to file:
" + e.getMessage());
        }
    }
}

public class VehicleServicingBillingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        List<Service> services = new ArrayList<>();
        services.add(new OilChange("Regular Oil Change", 500));
        services.add(new OilChange("Synthetic Oil Change", 800));
        services.add(new TireRotation("Standard Tire Rotation", 1100));
        services.add(new TireRotation("Balanced Tire Rotation", 900));
        services.add(new CarWash("Exterior Wash", 550));
        services.add(new CarWash("Interior and Exterior Wash", 1200));

        System.out.print("Enter owner's name: ");
        String ownerName = scanner.nextLine();
        System.out.print("Enter vehicle plate: ");
        String vehiclePlate = scanner.nextLine();

        VehicleServiceOrder serviceOrder = new VehicleServiceOrder(ownerName,
vehiclePlate);

        while (true) {
            System.out.println("Choose a service by entering its corresponding
number:");
            for (int i = 0; i < services.size(); i++) {
                System.out.println((i + 1) + ". " + services.get(i).name + " -
Rs." + services.get(i).price);
            }
            System.out.println("Enter '0' to finish selection.");
```

```
            System.out.print("Enter the index of the service: ");
            int serviceChoice = scanner.nextInt();
            scanner.nextLine();
            if (serviceChoice == 0) {
                break;
            }
            if (serviceChoice >= 1 && serviceChoice <= services.size()) {
                serviceOrder.addService(services.get(serviceChoice - 1));
            } else
             {
                System.out.println("Invalid service choice.");
            }
        }

        scanner.close();

        serviceOrder.display();

        System.out.println("Total Price: Rs." + serviceOrder.getTotalPrice());

        serviceOrder.saveToFile("service_bill.txt");
    }
}
```

Input/Output :

```
Enter owner's name: Sanskriti Kuwar
Enter vehicle plate: 789465
Choose a service by entering its corresponding number:
1. Regular Oil Change - Rs.500.0
2. Synthetic Oil Change - Rs.800.0
3. Standard Tire Rotation - Rs.1100.0
4. Balanced Tire Rotation - Rs.900.0
5. Exterior Wash - Rs.550.0
6. Interior and Exterior Wash - Rs.1200.0
Enter '0' to finish selection.
Enter the index of the service: 4
Choose a service by entering its corresponding number:
1. Regular Oil Change - Rs.500.0
2. Synthetic Oil Change - Rs.800.0
3. Standard Tire Rotation - Rs.1100.0
4. Balanced Tire Rotation - Rs.900.0
5. Exterior Wash - Rs.550.0
6. Interior and Exterior Wash - Rs.1200.0
Enter '0' to finish selection.
Enter the index of the service: 3
Choose a service by entering its corresponding number:
1. Regular Oil Change - Rs.500.0
2. Synthetic Oil Change - Rs.800.0
3. Standard Tire Rotation - Rs.1100.0
4. Balanced Tire Rotation - Rs.900.0
5. Exterior Wash - Rs.550.0
6. Interior and Exterior Wash - Rs.1200.0
Enter '0' to finish selection.
Enter the index of the service: 0
Service Order for Sanskriti Kuwar's Vehicle (Plate: 789465):
Tire Rotation Service: Balanced Tire Rotation - Rs.900.0
Tire Rotation Service: Standard Tire Rotation - Rs.1100.0
Total Price: Rs.2000.0
Service order details appended to service_bill.txt
```

Bill Generated :