NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS

CS-1004 Object Oriented Programming
Semester Project (Fall 2023)

Section (A and B)

Submission Information

Submission Deadline: **Wednesday 6th December, 2023 at 10:00 PM**. You are supposed to submit your assignment on GOOGLE CLASSROOM (CLASSROOM TAB not lab). Only ".ZIP" files are acceptable. Other formats should be directly given ZERO. Correct and timely submission of the assignment is the responsibility of every student, hence no relaxation will be given to anyone. Late Submission policy will be applied as described in course outline.

Tips: For timely completion of the assignment, start as early as possible.
Plagiarism: Plagiarism is not allowed. If found plagiarized, you will be awarded zero marks in the project and you will be awarded F grade in course.

Instructions:

Dear students, we will be using auto-grading tools, so failure to submit according to the format below would result in zero marks in the relevant evaluation instrument.

I. Implement separate .cpp and .h files for each class. Use a single main.cpp file to implement all your classes in your driver functions.
II. Combine all your work in one folder. The folder must contain only .cpp, .h and text files (no binaries, no exe files etc.).
III. Run and test your program on a lab machine before submission.
IV. Rename the folder as ROLL-NUM_SECTION (e.g. 22i-0001_A) and compress the folder as a zip file. (e.g. 22i-0001_A.zip). Do not submit .rar file.
V. Submit the .zip file on Google Classroom within the specified deadline.
VI. Submission other than Google classroom (e.g. email etc.) will not be accepted.
VII. The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.
VIII. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.
IX. Total Marks: 250.

Learning Objectives:

The AIM of Semester project is to hands-on file handling, composition, aggregation, inheritances and polymorphism (In short all concepts of OOP). We are also expecting you to implement basic concepts of cybersecurity (privacy, confidentiality, integrity and access control) in project. It is compulsory to write comments properly explaining every part of your code in your own words (Roman Urdu preferred), otherwise no marks will be awarded.

Warning:

- If there is a syntax error in the code, zero marks will be awarded in that part of the assignment.
- Your code must be generic and handle all errors and exceptions.

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS

CS-1004 Object Oriented Programming

Semester Project (Fall 2023)

Section (A and B)

In this project, we will develop a comprehensive application designed to manage and facilitate transactions related to cars. The system aims to streamline the process of buying, selling, and maintaining vehicles. Users, including car dealers and individual car owners, will benefit from the various features and functionalities provided by the application.

1. **Inventory Management:** Dealers can upload and manage their car inventory, including details like make, model, year, mileage, and price. Search and filter functionalities for users to easily find specific cars based on preferences.
2. **Sales and Purchase Module:** A user-friendly interface for buyers to browse available cars, view detailed specifications, and initiate purchase inquiries. Dealers can manage sales, update the status of cars (e.g., available, sold), and handle purchase requests.
3. **Maintenance Tracker**: Allows car owners to log and track maintenance activities, service schedules, and repairs for their vehicles. Dealers can provide maintenance history for cars in their inventory.
4. **Financing and Insurance Integration**: Integration with financial institutions for users to explore financing options. Information on insurance plans and options available for each vehicle.
5. **Appointment Scheduling**: Car owners can schedule appointments for test drives, inspections, or maintenance services through the application. Dealers can manage and confirm appointments, enhancing the overall customer experience.
6. **Document Management:** *Securely store* and manage important documents such as vehicle registration, insurance, and service records. Dealers can upload and share relevant documents with buyers during the transaction process.
7. **User Profiles and Authentication**: Secure user authentication to ensure data privacy and secure transactions. User profiles for both dealers and individual car owners, allowing them to track their activities within the application.

The system is supposed to have three basic users, Each with its own set of functionalities

1. **Buyer:**
    a. **Register:** Provide with the basic information of the user, which will be stored in a file.
    b. **Login:** The system reads the file to validate credentials, and allows access to the buyer to carry on with the application.
    c. **View vehicles:** All the vehicles will have a Name, company, model, mileage, price, description, seller name and contact, fuel type, engine, rating, make year, available in city name, and color. There are further 3 categories vehicles:
        i. Used Cars
        ii. New Cars
        iii. Bikes

Once chosen from the above category, the user can sort the list of available cars on the bases of company, price range, model, make year, vehicle name, and city. The information of the vehicles is stored on a file as well which is used to fetch and display the information. The user can select a specific vehicle, which will give the user the entire detail of that specific vehicle, and an option to contact the seller.

d. **Contact Seller**: User gets an option to contact the seller of a specific vehicle. The application displays simulation of calling, and a notification is added to that specific seller side with the information of the seller and the vehicle trying to purchase.
e. **View Auction:** User can view vehicles registered in the auction. All the vehicles will have a Name, company, model, mileage, price, description, seller name and contact, fuel type, engine, rating, make year, available in city name, and color. There are further 3 categories vehicles as well:
    i. Used Cars
    ii. New Cars
    iii. Bikes

    Once chosen from the above category, the user can sort the list of available cars on the bases of company, price range, model, make year, vehicle name, and city. The information of the vehicles is stored on a file as well which is used to fetch and display the information. The user can select a specific vehicle, which will give the user the entire detail of that specific vehicle, and an option to contact the seller.
f. **Request Inspection Report:** Once the user selects a vehicle, the application will provide with an option to get an inspection report, that is added by the admin in a file for the specific file.
g. **View Notifications**: Users can view notifications that are uploaded by the admin.

2. **Seller:**
    a. **Register:** Provide with the basic information of the user, which will be stored in a file.
    b. **Login:** The system reads the file to validate credentials, and allows access to the buyer to carry on with the application.
    c. **View Vehicles:** All the vehicles will have a Name, company, model, mileage, price, description, seller name and contact, fuel type, engine, rating, make year, available in city name, and color. There are further 3 categories vehicles:
        i. Used Cars
        ii. New Cars
        iii. Bikes

        Once chosen from the above category, the user can sort the list of available cars on the bases of company, price range, model, make year, vehicle name, and city. The information of the vehicles is stored on a file as well which is used to fetch and

display the information. The user can select a specific vehicle, which will give the user the entire detail of that specific vehicle, and an option to contact the seller.

d. **Add Vehicle:** Sellers can add new vehicles to the system by writing in the file saving the car information, provided with a check that makes sure the seller has filled all the required fields. As soon as the vehicle is added a request is sent to the admin who will generate an inspection report.
e. **Remove Vehicle:** Sellers can remove an existing vehicle from the system, that removes the information of that vehicle from the file. The system must check that that specific car is uploaded by that specific seller.
f. **Register In Auction**: Seller gets an option of adding a vehicle in auction. The seller must provide information about the vehicle that is to be auctioned.
g. **View Notifications**: User can view notifications that are uploaded by the admin.

3. **Admin:**
    a. **Login:** The system reads the file to validate credentials, and allows access to the admin to carry on with the application.
    b. **Add Vehicle**: Admin can add vehicles by themselves, by writing into the file. The admin will have to fill all the details required for adding a vehicle.
    c. **Remove Vehicle**: Admin can remove a vehicle just like a seller, by updating the file.
    d. **Manage Inspection Requests:** Once a car is added, the admin will get a request to generate an inspection report for that specific file. The report will include Car Detail: Customer/Dealer Name, Engine-capacity, Mileage, Transmission Type, Inspection Date, Chassis No, Engine-no, Registration No, Fuel-type, Petrol, Color, Location, Registered City, Current City, Car features, rating of the vehicle.
    e. **Add Notification:** Admin can add notifications about the system that can be viewable by seller and buyer.
    f. **Remove Notification**: Admin can remove a notification uploaded.
    g. **Rate a Vehicle**: Admin can rate a vehicle on the basis of inspection that will be visible in the view vehicle option.

You are tasked to identify and create separate classes for the entities, and implement it. Draw a class diagram for the identified classes with proper identification of the relationships. Add appropriate functionalities, as mentioned in the roles of the user. You are required to identify appropriate relationship between different classes and apply polymorphic behavior between them. Furthermore, you are required to add a secure mechanism of your choice taught in the course of “network and cyber security”, to secure your files (text or binary). A proper menu driven is required from your end.

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS

CS-1004 Object Oriented Programming
Semester Project (Fall 2023)

Section (A and B)

Add as much detail and functionality to the project on your own as needed. Your implementation should demonstrate file handling, operator overloading, polymorphism and virtual functions, as well as all other concepts taught in the course. Output should be clear and easy to follow. Write your driver program such that you can easily demonstrate all features. Bonus marks will be awarded if you implement any features extraordinarily well and thoroughly, and for original thought and design.

Menu-Driven Interface:

Create a menu structure for each user type once successfully logged in:

Buyer Menu:

1. View Vehicles
2. View Auction
3. Request Inspection Report
4. Contact Seller
5. View Notifications
6. Add a comment on car
7. Logout

Seller Menu:

1. View Vehicles
2. Add Vehicle
3. Remove Vehicle
4. Register in Auction
5. View Notifications
6. Logout

Admin Menu:

1. Add Vehicle
2. Remove Vehicle
3. Manage Inspection Requests
4. Add Notification
5. Remove Notification and comments
6. Rate a Vehicle
7. Logout

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS

CS-1004 Object Oriented Programming
Semester Project (Fall 2023)

Section (A and B)

Testing Plan:

- Test each class's methods independently to ensure they perform as expected. Verify that file handling operations (read/write) work correctly.
- Test the interactions between different classes, ensuring seamless communication. Verify that user authentication and authorization mechanisms are effective.
- Test each functionality described in the document, such as adding/removing vehicles, viewing notifications, scheduling appointments, etc.
- Validate the security measures implemented for file handling. Check the robustness of the user authentication system.
- Ensure the menu-driven interface is user-friendly. Validate that users can easily navigate through the system.

Marking Rubric:

Item	Description	Marks
Appropriate design	1. Class diagram is appropriate for the problem. **(10 points)** 2. Maps to the code **(5 points)** 3. Does not violate OOP principles of encapsulation, modularity, abstraction **(10 points)** 4. Correctly implements relationships. **(10 points)** **Total:** 35 point	
Seller Module	1. Secure login and registration **(5 points)** 2. Validate credentials against stored user information **(5 points)** 3. Allow buyers to view vehicles based on categories and sort options **(10 points)** 4. Display detailed information about selected vehicles. **(5 points)** 5. Contact Seller **(5 points)** 6. View Auction **(5 points)** 7. Request Inspection Report **(5 points)** 8. View Notifications **(5 points)** **Total:** 45 points	
Buyer Module	1. Secure login and registration (5 points) 2. Validate credentials against stored user information	

CS-1004 Object Oriented Programming
Semester Project (Fall 2023)

Section (A and B)

	(5 points) 3. View Vehicles **(10 points)** 4. Add vehicle **(5 points)** 5. Remove Vehicle **(5 points)** 6. Register in Auction **(5 points)** 7. View Notifications **(5 points)** **Total:** 40 points	
Admin Module	1. Login **(5 points)** 2. Add Vehicle **(5 points)** 3. Remove Vehicle **(5 points)** 4. Manage Inspection Requests **(5 points)** 5. Add/Remove Notifications and Rate a Vehicle **(5 points)** **Total:** 25 points	
File handling	1. Proper file handling mechanisms should be implemented for storing user, vehicle, and other relevant information. **(10 points)** 2. Security Implementation **(10 points)** 3. Updating files **(10 points)** **Total:** 30 points	
Sufficient data pre populated for testing program	Develop and execute a comprehensive testing plan covering all functionalities. Ensure robustness and correctness of the implemented features. **(10 points)** **Total:** 10 points	
Polymorphism	Implementing complete polymorphic concept. **(15 points)**	
Dynamic memory allocation	Using dynamic arrays or objects appropriately where required. **(20 points)**	
Communication	Communication between all components is appropriately simulated and does not violate OOP principles (do not use global functions, public data members or too many friend functions) **(15 points)** **Total:** 15 points	
Driver program and output	1. marks for implemented code that is not demonstrated in the output. **(5 points)** 2. Good menu design **(5 points)** 3. Easy user interaction. **(5 points)**	

	Total: 15 points	
	Grand Total: 250 points	
Bonus points	1. Hide Passwords **(3 points)** 2. Appointment scheduling **(3 points)** 3. Document management **(3 points)** 4. Interactive simulation **(3 points)** 5. Performance optimization **(3 points)**	

Submission Instructions:

- Combine all separate files into a single zip file before submitting.
- Your code MUST be in separate .cpp and .h files for each class, and a .cpp file called driver.cpp that is the driver code.
- Submit your code on Google Classroom with the following naming convention: Rollnumber1.zip