Ride Sharing System: OOP in Smalltalk and C++

Introduction

This report examines a Ride Sharing System implemented in Smalltalk and C++, showcasing encapsulation, inheritance, and polymorphism. The system includes Ride, StandardRide, PremiumRide, Driver, and Rider classes to manage rides, fares, and user data. Code is available at https://github.com/mohitmurali/632_AdvancedProgramming/tree/main/Assignment-5 .

Encapsulation

Encapsulation protects data by restricting access to defined methods, ensuring security (Stroustrup, 2013). In C++, Ride marks fare as protected, accessible to subclasses (StandardRide, PremiumRide) via getFare() and setFare(). Driver and Rider keep assignedTrips and bookedTrips private, using addRide() and requestRide() for access, preventing external changes.

In Smalltalk, instance variables like fare in Ride are private by default, accessed through fare and setFare:. Driver's assignedTrips and Rider's bookedTrips are modified only via addRide: and requestRide:. Both languages enforce encapsulation, with Smalltalk's dynamic typing simplifying control compared to C++'s explicit specifiers.

Inheritance

Inheritance enables code reuse by extending a base class (Gamma et al., 1994). In C++, StandardRide and PremiumRide inherit from Ride publicly, reusing rideCode and distance while overriding calculateFare() to set fares at $1.5 and $3.0 per mile, respectively.

In Smalltalk, StandardRide and PremiumRide subclass Ride, inheriting showRideDetails and overriding calculateFare with the same pricing. Both implementations leverage inheritance to share Ride's functionality, allowing specialized fare logic.

Polymorphism

Polymorphism lets objects of different types respond uniquely to the same interface (Meyer, 1997). In C++, a std::vector<Ride*> holds StandardRide and PremiumRide objects. Iterating in main() calls calculateFare() and showRideDetails() virtually, executing subclass-specific methods.

In Smalltalk, an OrderedCollection stores rides. The test script iterates with do:, sending calculateFare and showRideDetails, resolved at runtime by object type. C++ uses virtual functions, while Smalltalk's dynamic typing ensures flexible polymorphism.

Conclusion

The Ride Sharing System demonstrates encapsulation, inheritance, and polymorphism effectively in Smalltalk and C++, meeting OOP requirements through protected data, reused code, and dynamic behavior.

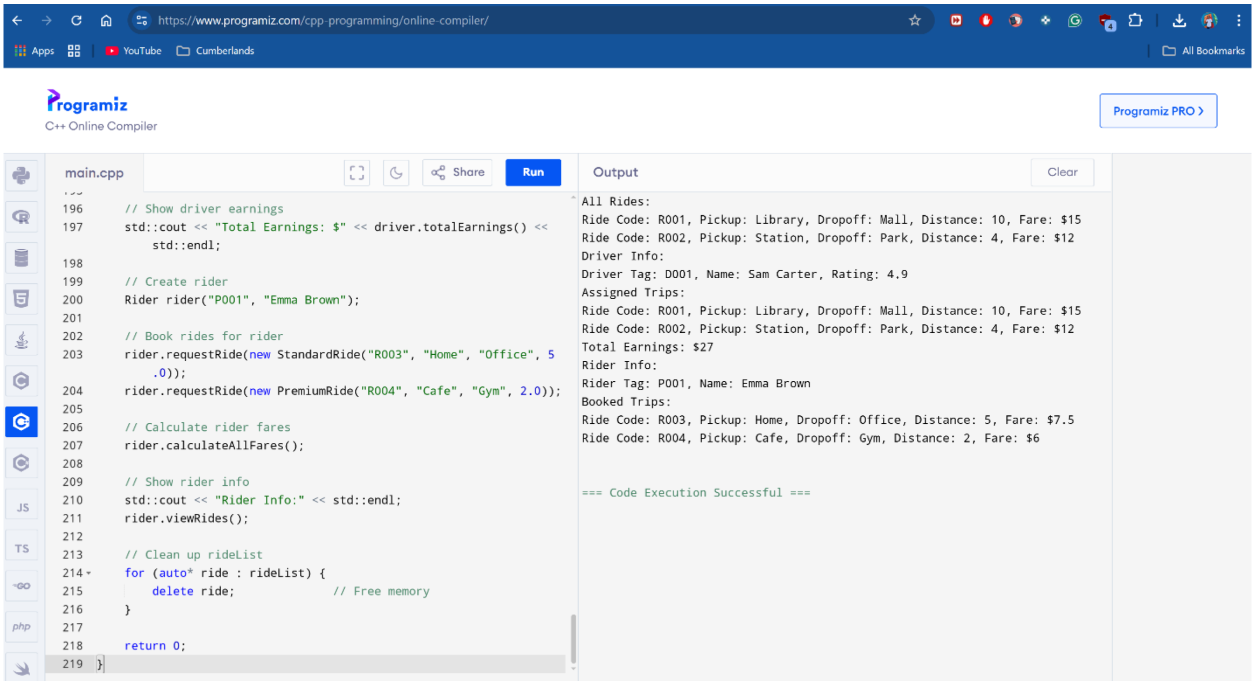
References

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns*. Addison-Wesley.

Meyer, B. (1997). *Object-oriented software construction* (2nd ed.). Prentice Hall.

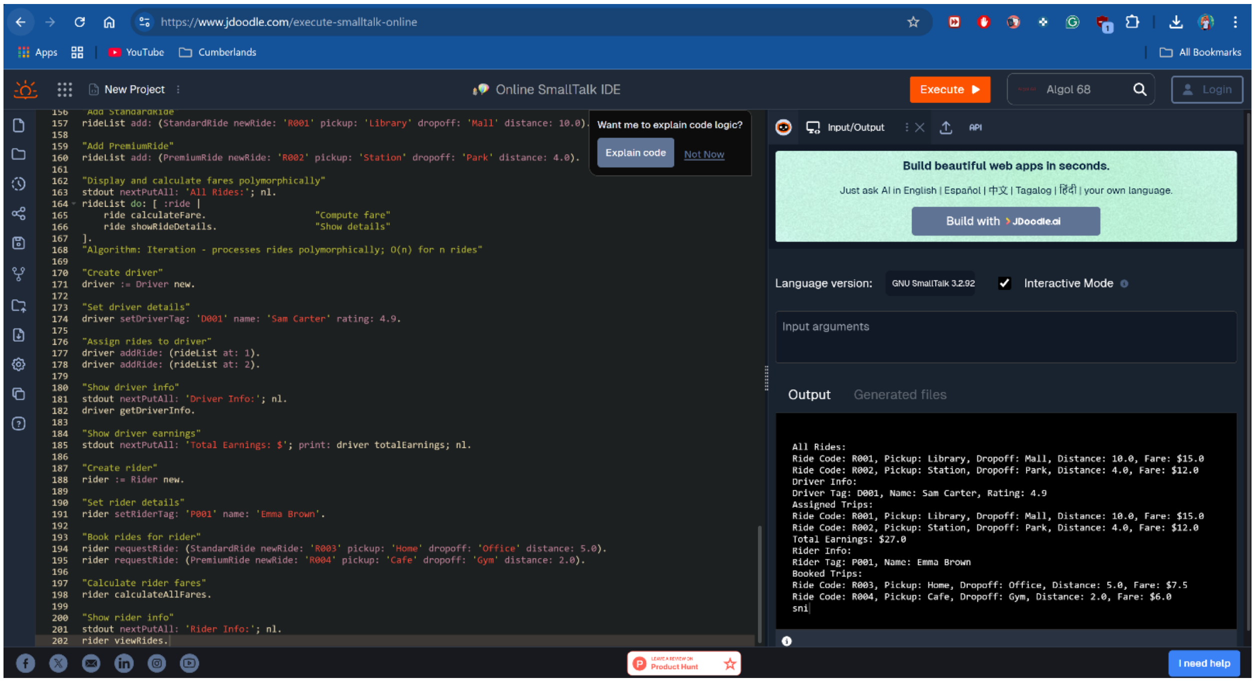
Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley.

Screenshots of outputs



```cpp
    // Show driver earnings
    std::cout << "Total Earnings: $" << driver.totalEarnings() <<
        std::endl;

    // Create rider
    Rider rider("P001", "Emma Brown");

    // Book rides for rider
    rider.requestRide(new StandardRide("R003", "Home", "Office", 5
        .0));
    rider.requestRide(new PremiumRide("R004", "Cafe", "Gym", 2.0));

    // Calculate rider fares
    rider.calculateAllFares();

    // Show rider info
    std::cout << "Rider Info:" << std::endl;
    rider.viewRides();

    // Clean up rideList
    for (auto* ride : rideList) {
        delete ride;                // Free memory
    }

    return 0;
}
```

```
All Rides:
Ride Code: R001, Pickup: Library, Dropoff: Mall, Distance: 10, Fare: $15
Ride Code: R002, Pickup: Station, Dropoff: Park, Distance: 4, Fare: $12
Driver Info:
Driver Tag: D001, Name: Sam Carter, Rating: 4.9
Assigned Trips:
Ride Code: R001, Pickup: Library, Dropoff: Mall, Distance: 10, Fare: $15
Ride Code: R002, Pickup: Station, Dropoff: Park, Distance: 4, Fare: $12
Total Earnings: $27
Rider Info:
Rider Tag: P001, Name: Emma Brown
Booked Trips:
Ride Code: R003, Pickup: Home, Dropoff: Office, Distance: 5, Fare: $7.5
Ride Code: R004, Pickup: Cafe, Dropoff: Gym, Distance: 2, Fare: $6


=== Code Execution Successful ===
```


https://www.jdoodle.com/execute-smalltalk-online
Apps
YouTube
Cumberlands
All Bookmarks
New Project
Online SmallTalk IDE
Execute
Algol 68
Login
Want me to explain code logic?
Explain code
Not Now
Input/Output
API
Build beautiful web apps in seconds.
Just ask AI in English | Español | 中文 | Tagalog | हिंदी | your own language.
Build with JDoodle.ai
Input arguments
Output
Generated files
I need help

Language version: GNU SmallTalk 3.2.92 ☑ Interactive Mode

```smalltalk
"Add StandardRide"
rideList add: (StandardRide newRide: 'R001' pickup: 'Library' dropoff: 'Mall' distance: 10.0).

"Add PremiumRide"
rideList add: (PremiumRide newRide: 'R002' pickup: 'Station' dropoff: 'Park' distance: 4.0).

"Display and calculate fares polymorphically"
stdout nextPutAll: 'All Rides:'; nl.
rideList do: [ :ride |
    ride calculateFare.                     "Compute fare"
    ride showRideDetails.                   "Show details"
].
"Algorithm: Iteration - processes rides polymorphically; O(n) for n rides"

"Create driver"
driver := Driver new.

"Set driver details"
driver setDriverTag: 'D001' name: 'Sam Carter' rating: 4.9.

"Assign rides to driver"
driver addRide: (rideList at: 1).
driver addRide: (rideList at: 2).

"Show driver info"
stdout nextPutAll: 'Driver Info:'; nl.
driver getDriverInfo.

"Show driver earnings"
stdout nextPutAll: 'Total Earnings: $'; print: driver totalEarnings; nl.

"Create rider"
rider := Rider new.

"Set rider details"
rider setRiderTag: 'P001' name: 'Emma Brown'.

"Book rides for rider"
rider requestRide: (StandardRide newRide: 'R003' pickup: 'Home' dropoff: 'Office' distance: 5.0).
rider requestRide: (PremiumRide newRide: 'R004' pickup: 'Cafe' dropoff: 'Gym' distance: 2.0).

"Calculate rider fares"
rider calculateAllFares.

"Show rider info"
stdout nextPutAll: 'Rider Info:'; nl.
rider viewRides.
```

```
All Rides:
Ride Code: R001, Pickup: Library, Dropoff: Mall, Distance: 10.0, Fare: $15.0
Ride Code: R002, Pickup: Station, Dropoff: Park, Distance: 4.0, Fare: $12.0
Driver Info:
Driver Tag: D001, Name: Sam Carter, Rating: 4.9
Assigned Trips:
Ride Code: R001, Pickup: Library, Dropoff: Mall, Distance: 10.0, Fare: $15.0
Ride Code: R002, Pickup: Station, Dropoff: Park, Distance: 4.0, Fare: $12.0
Total Earnings: $27.0
Rider Info:
Rider Tag: P001, Name: Emma Brown
Booked Trips:
Ride Code: R003, Pickup: Home, Dropoff: Office, Distance: 5.0, Fare: $7.5
Ride Code: R004, Pickup: Cafe, Dropoff: Gym, Distance: 2.0, Fare: $6.0
sni
```