Assignment 5: Developing a Class-Based Ride Sharing System

Haeri Kyoung

University of the Cumberlands

MSCS-632-A01 – Advanced Programming Language

Professor Jay Thom

Sep 20, 2025

Overview

This assignment required the development of a class-based Ride Sharing System in two languages, C++ and Smalltalk. The goal was to demonstrate three fundamental object-oriented programming principles: encapsulation, inheritance, and polymorphism. The project included a base Ride class, two ride subclasses (StandardRide and PremiumRide), as well as supporting Driver and Rider classes. Both implementations produced sample output that showed the system's ability to calculate fares, track driver earnings, and display rider history.

Encapsulation

In the C++ implementation, encapsulation was achieved by declaring the assignedRides and requestedRides vectors as private members of the Driver and Rider classes. They were accessible only through defined methods such as addRide() and requestRide(). This ensured that rides could not be manipulated directly outside of the class, enforcing controlled access.
In Smalltalk, encapsulation is inherent because instance variables are not directly accessible from outside the class. The Driver class maintained an internal assignedRides collection, and the Rider class kept a requestedRides collection. Access and modification occurred only through methods like addRide: and requestRide:.

Inheritance

The Ride class served as the base in both languages. Subclasses StandardRide and PremiumRide inherited from it and specialized the fare calculation logic. In C++, this was done using virtual methods overridden in the subclasses. In Smalltalk, subclassing was implemented by creating StandardRide and PremiumRide as subclasses of Ride and overriding the fare method. This

design illustrated how common attributes and behaviors could be reused while allowing each subclass to implement its own variation.

Polymorphism

Polymorphism was demonstrated by storing rides of different types in a single collection and invoking methods dynamically. In C++, a vector of shared_ptr<Ride> held both standard and premium rides. Calling fare() or rideDetails() on this collection resulted in the appropriate subclass method executing. In Smalltalk, an OrderedCollection stored different ride objects, and sending the fare or rideDetails message to each element correctly dispatched to the right subclass implementation.

Observations

Developing the system in C++ was straightforward because of familiarity with the syntax, strong typing, and standard libraries. The main challenge was ensuring memory safety and using smart pointers (shared_ptr) to avoid manual memory management. In contrast, Smalltalk presented a steeper learning curve due to its purely object-oriented model and unique development environment. However, its uniformity where everything is an object and all computation happens through message passing was conceptually elegant. The live environment also made it easy to experiment and view results immediately.

Overall, the assignment reinforced how different languages implement core OOP principles. C++ showed a practical and widely used approach with strong typing and compiler support, while Smalltalk highlighted the conceptual purity of object orientation.

GitHub Repository

https://github.com/hkyoung38554/MSCS632_Assignment5

Screenshots

```
haerikyoung@Mac cpp % cd "/Users/haerikyoung/Desktop/Doc/UC/MSCS632_Assignment5/cpp"
./rides | tee sample_output_cpp.txt

cd: no such file or directory: /Users/haerikyoung/Desktop/Doc/UC/MSCS632_Assignment5/cpp
=== Polymorphic ride listing ===
[Standard] R-1001 Airport -> Downtown | distance: 12.4 mi | fare: $20.60
[Premium] R-1002 Campus -> Museum | distance: 6.10 mi | fare: $24.33
[Standard] R-1003 Stadium -> Hotel | distance: 3.80 mi | fare: $7.70

=== Driver details ===
Driver Alex (D-77), rating 4.94
Completed rides: 3
[Standard] R-1001 Airport -> Downtown | distance: 12.40 mi | fare: $20.60
[Premium] R-1002 Campus -> Museum | distance: 6.10 mi | fare: $24.33
[Standard] R-1003 Stadium -> Hotel | distance: 3.80 mi | fare: $7.70
Total earnings: $52.63

=== Rider history ===
Rider Haeri (U-55) ride history: 2 rides
[Standard] R-1001 Airport -> Downtown | distance: 12.40 mi | fare: $20.60
[Standard] R-1003 Stadium -> Hotel | distance: 3.80 mi | fare: $7.70
```

```
haerikyoung@Mac MSCS632_Assignment5 % cd smalltalk
gst rides.st | tee sample_output_smalltalk.txt

Object: nil error: did not understand #add:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #add: (SysExcept.st:1448)
Driver>>addRide: (rides.st:71)
UndefinedObject>>executeStatements (rides.st:117)
Object: nil error: did not understand #add:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #add: (SysExcept.st:1448)
Driver>>addRide: (rides.st:71)
UndefinedObject>>executeStatements (rides.st:117)
Object: nil error: did not understand #add:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #add: (SysExcept.st:"Global ga1448rbage collection... done"
)
Driver>>addRide: (rides.st:71)
UndefinedObject>>executeStatements (rides.st:117)
Object: nil error: did not understand #add:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #add: (SysExcept.st:1448)
Rider>>requestRide: (rides.st:100)
UndefinedObject>>executeStatements (rides.st:120)
Object: nil error: did not understand #add:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #add: (SysExcept.st:1448)
Rider>>requestRide: (rides.st:100)
UndefinedObject>>executeStatements (rides.st:120)
=== Polymorphic ride listing ===
[Standard] R-1001 Airport -> DowntownObject: 1 error: The program attempted to divide a number by zero
ZeroDivide(Exception)>>signal (ExcHandling.st:254)
SmallInteger(Number)>>zeroDivide (SysExcept.st:1426)
Fraction>>setNumerator:setDenominator: (Fraction.st:485)
Fraction class>>numerator:denominator: (Fraction.st:66)
Fraction>>- (Fraction.st:151)
FloatD(Float)>>printOn:special: (Float.st:533)
FloatD(Float)>>printOn: (Float.st:436)
FloatD(Object)>>printString (Object.st:534)
StandardRide(Ride)>>rideDetails (rides.st:27)
optimized [] in UndefinedObject>>executeStatements (rides.st:123)
OrderedCollection>>do: (OrderColl.st:67)
UndefinedObject>>executeStatements (rides.st:123)

=== Driver details ===
Object: 1 error: The program attempted to divide a number by zero
ZeroDivide(Exception)>>signal (ExcHandling.st:254)
SmallInteger(Number)>>zeroDivide (SysExcept.st:1426)
Fraction>>setNumerator:setDenominator: (Fraction.st:485)
Fraction class>>numerator:denominator: (Fraction.st:66)
Fraction>>- (Fraction.st:151)
FloatD(Float)>>printOn:special: (Float.st:533)
FloatD(Float)>>printOn: (Float.st:436)
FloatD(Object)>>printString (Object.st:534)
Driver>>getDriverInfo (rides.st:74)
UndefinedObject>>"Global garexecuteStatementsbage collection... done, heap grown"
 (rides.st:126)
Object: nil error: did not understand #do:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #do: (SysExcept.st:1448)
Driver>>listRides (rides.st:81)
UndefinedObject>>executeStatements (rides.st:126)

=== Rider history ===
Rider Haeri (U-55) ride history: 0 rides
Object: nil error: did not understand #do:
MessageNotUnderstood(Exception)>>signal (ExcHandling.st:254)
UndefinedObject(Object)>>doesNotUnderstand: #do: (SysExcept.st:1448)
Rider>>viewRides (rides.st:104)
UndefinedObject>>executeStatements (rides.st:129)
```

References

Goldberg, A., & Robson, D. (1983). Smalltalk-80: The Language and Its Implementation. Addison-Wesley.

Sebesta, R. W. (2020). Concepts of Programming Languages (12th ed.). Pearson.