**Assignment 5: Developing a Class-Based Ride Sharing System**

Shrisan Kapali

Advanced Programming Languages (MSCS 632 M20)

Ulrich Vouama  
July 27, 2025

**GitHub Link**

[**https://github.com/ShrisanKapali-Cumberlands/MSCS\_632\_Assignment\_5**](https://github.com/ShrisanKapali-Cumberlands/MSCS_632_Assignment_5)

**C++ Ride Sharing Program**

The C++ Program implements a ride-sharing system by applying fundamental principles of the OOP.

Encapsulation is used in the Driver and Rider classes, where the assigned rides and requested rides are declared as private.

**A screenshot of a computer program

AI-generated content may be incorrect. A computer code with black text

AI-generated content may be incorrect.**

Both the standard ride and premium ride classes inherit the base Ride class and acquire properties of the base Ride class such as rideDetails() and getFare().

A close up of a text

AI-generated content may be incorrect. A close up of text

AI-generated content may be incorrect.

The allRides vector class pointers demonstrate polymorphism by storing different ride types in a single list of collection. Based on the ride type, the calculateFare() method is determined during the runtime.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

The use of pure virtual method in the Ride base class demonstrates the use of abstraction.

A screenshot of a computer

AI-generated content may be incorrect.

**Program Output**

**A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect. A screen shot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Cases**

**Case 1: Multiple Rides for a driver and rider**

A computer code on a white background

AI-generated content may be incorrect.

**Output**

**A screenshot of a computer screen

AI-generated content may be incorrect. A screen shot of a black screen

AI-generated content may be incorrect.**

**Case 2: Single Ride for a driver and rider**

**A screen shot of a computer

AI-generated content may be incorrect.** **A screen shot of a black screen

AI-generated content may be incorrect.**

**Case 3: No Ride for a driver and rider**

**A black background with white text

AI-generated content may be incorrect. A black and white text on a black background

AI-generated content may be incorrect.**

**Smalltalk Ride Sharing Program**

The program was compiled and run using Pharo Launcher.

**Base Ride Class**

A screenshot of a computer

AI-generated content may be incorrect.

The Premium and Standard Ride Classes inherited the Ride Class, and these classes override the base calculateFare method. A screenshot of a computer

AI-generated content may be incorrect.

The Drive and Rider Class have private instance variables with access methods which demonstrate encapsulation.

A screenshot of a computer

AI-generated content may be incorrect.

The different fare calculations for different ride types demonstrate polymorphism.

**Example Test Program with Output**  
A screenshot of a computer

AI-generated content may be incorrect.