

## Week 6

The screenshot shows a Java code editor with four tabs open, each displaying a different class from the RIDEAPP project:

- RideSelector.java**: Contains a static method `selectRide` that takes a distance and returns an instance of `Ride`. It uses a switch statement to determine which ride type to return based on the distance.
- Ride.java**: A public interface with two methods: `calculateFare(double distance)` and `getRideType()`.
- Prime.java**: A concrete implementation of the `Ride` interface. It has a static final variable `RATE_PER_KM` set to 15.0. It overrides the `calculateFare` method to return `RATE_PER_KM * distance` and the `getRideType` method to return "prime".
- Mini.java**: Another concrete implementation of the `Ride` interface. It has a static final variable `RATE_PER_KM` set to 10.0. It overrides the `calculateFare` method to return `RATE_PER_KM * distance` and the `getRideType` method to return "mini".
- SUV.java**: A third concrete implementation of the `Ride` interface. It has a static final variable `RATE_PER_KM` set to 20.0. It overrides the `calculateFare` method to return `RATE_PER_KM * distance` and the `getRideType` method to return "suv".

Project Explorer:

- RIDEAPP
  - Minijava
  - Prime.java
  - Ride.java
  - RideSelector.java** 4
  - SUV.java
- > OUTLINE
- > TIMELINE
- > PROJECTS
- > RUN CONFIGURATION
- > JAVA PROJECTS
  - RIDEAPP
    - Minijava
    - Prime.java
    - Ride.java
    - RideSelector.java** 4
    - SUV.java

---

Code Editor:

```

RideSelector.java > Java > RideSelector > main(String[] args)
1 import java.util.Scanner;
2
3 public class RideSelector {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.println("Enter ride type (mini/prime/suv):");
9         String type = sc.nextLine().trim().toLowerCase(); // convert input to lowercase
10
11        System.out.println("Enter distance (km):");
12        double distance = sc.nextDouble();
13
14        Ride ride = null;
15
16        // match lowercase input
17        switch (type) {
18            case "mini":
19                ride = new Mini();
20                break;
21            case "prime":
22                ride = new Prime();
23                break;
24            case "suv":
25                ride = new SUV();
26                break;
27            default:
28                System.out.println("Invalid ride type!");
29                System.exit(status: 0);
30
31        double fare = ride.calculateFare(distance);
32        System.out.println("Ride Type: " + ride.getRideType());
33        System.out.println("Total Fare: ₹" + fare);
34
35        sc.close();
36    }
37 }
38

```

---

Terminal Output:

```

PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA\RideApp> cd; 
nts\Desktop\OOP JAVA\RideApp'; & 'C:\Program Files\Eclipse Adoptium\j
XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\VLT-BNT\AppDa
\c0cbe5ab50d5f01386298f4ecc04f8c2\redhat.java\jdt_ws\RideApp_ce3adf2c
Enter ride type (mini/prime/suv):
prime
Enter distance (km):
3
Ride Type: prime
Total Fare: ₹45.0
PS C:\Users\VLT-BNT\OneDrive\Documents\Desktop\OOP JAVA\RideApp>

```