Questions from Java OOPs Assignment:Problem Statement:The Department of Highways is installing a toll collection system on a major road. Trucks pay $5 per axle plus $10 per half-ton of total weight. The tollbooth displays total receipts and truck counts since the last collection.Create and compile the two interfaces:TruckTollBoothCreate two classes implementing the Truck interface:Each named after a truck make of your choice (two different makes)Include member variables for axles, total weight, and makeChoose appropriate access level modifiers considering which variables can changeProvide access methods but prevent modification of attributes externallyCreate a class implementing the TollBooth interface:Example: AlleghenyTollBooth
This class tracks:The total number of trucks that have gone throughThe total toll receipts since last collectionThe class should:Display the totals (receipts and trucks) since last collectionReset totals to zero upon receipt collection (simulating cash drawer removal)Simulate the tollbooth using a main class:Demonstrate the arrival of several trucks with different axles and weightsCall methods to calculate tolls, display totals, and collect receiptsConfirm your output matches the sample usage scenario given in the problem statement.

```
public interface Truck {
  public int getAxles();
  public int getTotalWeight();
  public String getMake();
}


public class FordTruck implements Truck {
  private final int axles;
  private final int totalWeight;
  private final String make;


  public FordTruck(int axles, int totalWeight) {
     this.axles = axles;
     this.totalWeight = totalWeight;
     this.make = "Ford";
  }


  public int getAxles() {
     return axles;
  }


  public int getTotalWeight() {
     return totalWeight;
  }


  public String getMake() {
     return make;
  }
}
```

```
public class NissanTruck implements Truck {
  private final int axles;
  private final int totalWeight;
  private final String make;


  public NissanTruck(int axles, int totalWeight) {
     this.axles = axles;
     this.totalWeight = totalWeight;
     this.make = "Nissan";
  }


  public int getAxles() {
     return axles;
  }


  public int getTotalWeight() {
     return totalWeight;
  }


  public String getMake() {
     return make;
  }
}


public class DaewooTruck implements Truck {
  private final int axles;
  private final int totalWeight;
  private final String make;


  public DaewooTruck(int axles, int totalWeight) {
     this.axles = axles;
     this.totalWeight = totalWeight;
     this.make = "Daewoo";
  }


  public int getAxles() {
     return axles;
  }


  public int getTotalWeight() {
     return totalWeight;
  }


  public String getMake() {
```

```
    return make;
  }
}


public interface TollBooth {
  public void calculateToll(Truck truck);
  public void displayData();
  public void collectReceipts();
}


public class AlleghenyTollBooth implements TollBooth {
  private int totalTrucks;
  private int totalReceipts;


  public AlleghenyTollBooth() {
    totalTrucks = 0;
    totalReceipts = 0;
  }


  public void calculateToll(Truck truck) {
    int axles = truck.getAxles();
    int weight = truck.getTotalWeight();
    int toll = (axles * 5) + ((weight / 500) * 10);
    totalTrucks++;
    totalReceipts += toll;
    System.out.println("Arrival of " + truck.getMake() + " Truck");
    System.out.println("Truck arrival - Axles: " + axles +
       " Total weight: " + weight + " Toll due: $" + toll);
  }


  public void displayData() {
    System.out.println("Totals since last collection - Receipts: $" + totalReceipts + " Trucks: " +
totalTrucks);
  }


  public void collectReceipts() {
    System.out.println("*** Collecting receipts ***");
    displayData();
    totalTrucks = 0;
    totalReceipts = 0;
    System.out.println("***** Reset Receipts **********");
    displayData();
  }
}


public class TestTollBooth {
```

```
  public static void main(String[] args) {
    TollBooth booth = new AlleghenyTollBooth();


    Truck ford = new FordTruck(5, 12500); // 5 axles, 12500 kg
    Truck nissan = new NissanTruck(2, 5000); // 2 axles, 5000 kg
    Truck daewoo = new DaewooTruck(6, 17000); // 6 axles, 17000 kg


    booth.calculateToll(ford);
    booth.calculateToll(nissan);
    booth.calculateToll(daewoo);


    booth.displayData();
    booth.collectReceipts();
  }
}
```