Isaac Ray Shoebottom
CS 1083
Assignment 1
3429069

Section A

Source Code:

```
/**
 This class is the vehicle object.
 @author Isaac Shoebottom (3429069)
 */

public class Vehicle {
    private static long uniqueIDTracker = 100;
    private final long uniqueID;
    private int numberOfPassengers;
    private final int axles;
    private final double weight;

    Vehicle(int numberOfPassengers, int wheels,double weight ) {
        this.weight = weight;
        this.axles = (wheels > 2) ? wheels/2 : 0;
        this.numberOfPassengers = numberOfPassengers;
        uniqueID = uniqueIDTracker + 1;
        uniqueIDTracker++;
    }


    public double getWeight() {
        return weight;
    }
    public int getAxles() {
        return axles;
    }
    public int getNumberOfPassengers() {
```

```
        return numberOfPassengers;
    }
    public long getUniqueID() {
        return uniqueID;
    }
    public void setNumberOfPassengers(int numberOfPassengers) {
        this.numberOfPassengers = numberOfPassengers;
    }

}
```

Section B

Source Code:

```
/**
 This class is the ferry object.
 @author Isaac Shoebottom (3429069)
 */



import java.text.DecimalFormat;
import java.util.Arrays;
import java.util.Objects;


public class Ferry {
    int maximumNumberOfVehicles;
    int maximumNumberOfAxles;
    int maximumNumberOfPassengers;
    double maxWeight;
    int numberOfVehicles;
    int numberOfAxles;
    int numberOfPassengers;
    double weight;
    Vehicle[] vehicles = new Vehicle[0];



    Ferry(int maximumNumberOfVehicles, int maximumNumberOfAxles, int
maximumNumberOfPassengers, double maxWeight) {
        this.maximumNumberOfVehicles = maximumNumberOfVehicles;
        this.maximumNumberOfAxles = maximumNumberOfAxles;
        this.maximumNumberOfPassengers = maximumNumberOfPassengers;
```

```
        this.maxWeight = maxWeight;
    }

    public boolean addVehicle(Vehicle newVehicle) {
        for(Vehicle vehicle: vehicles) {
            if (newVehicle.getUniqueID() == vehicle.getUniqueID()) {
                return false;
            }
        }
        if (newVehicle.getWeight() + weight > maxWeight ||
                newVehicle.getAxles() +numberOfAxles >
maximumNumberOfAxles ||
                newVehicle.getNumberOfPassengers() +
numberOfPassengers > maximumNumberOfPassengers ||
                1+numberOfVehicles > maximumNumberOfVehicles)
        {
            return false;
        }
        int counter = vehicles.length;
        vehicles = Arrays.copyOf(vehicles, vehicles.length + 1);
        vehicles[counter] = newVehicle;
        addOrRemoveSpecs(newVehicle, 1);
        return true;
    }
    /*
    public boolean addVehicles(Vehicle... newVehicles) {
        double[] vehicleInfo =
calculateTotalSpecsForArray(newVehicles);
        //axles = 1, passengers = 2, weight = 0, vehicles = 3
        if (vehicleInfo[0] > maxWeight ||
                vehicleInfo[1] > maximumNumberOfAxles ||
```

```
                vehicleInfo[2] > maximumNumberOfPassengers ||
                vehicleInfo[3] > maximumNumberOfVehicles ||
                doIDsMatch(vehicles, newVehicles))
        {
            return false;
        }
        addOrRemoveSpecs(newVehicles, 1);
        int counter = vehicles.length;
        vehicles = Arrays.copyOf(vehicles, vehicles.length +
newVehicles.length);
        for (Vehicle vehicle: newVehicles) {
            vehicles[counter] = vehicle;
        }
        return true;
    }
     */
    public boolean removeVehicle(Vehicle newVehicle) {
        for (int i = 0; i < vehicles.length; i++) {
            if (vehicles[i].getUniqueID() == newVehicle.getUniqueID())
{
                vehicles[i] = null;
                addOrRemoveSpecs(newVehicle, -1);
                vehicles =
Arrays.stream(vehicles).filter(Objects::nonNull).toArray(Vehicle[]::ne
w);
                return true;
            }
        }
        return false;
    }


    public void addOrRemoveSpecs(Vehicle vehicle, int sign) {
```

```
        numberOfVehicles += sign;
        numberOfAxles += sign*vehicle.getAxles();
        numberOfPassengers += sign*vehicle.getNumberOfPassengers();
        weight += sign*vehicle.getWeight();
    }

    /*
    private double[] calculateTotalSpecsForArray(Vehicle[]
tempVehicles) {
        double totalWeight = 0;
        int totalAxles = 0;
        int totalPassengers = 0;
        int totalNumberOfCars = tempVehicles.length;
        for (Vehicle vehicle: tempVehicles) {
            totalWeight += vehicle.getWeight();
            totalAxles += vehicle.getAxles();
            totalPassengers += vehicle.getNumberOfPassengers();
        }
        return new double[]{totalWeight, totalAxles, totalPassengers,
totalNumberOfCars};
    }

    private void addOrRemoveSpecs(Vehicle[] tempVehicles, int sign) {
        double[] tempSpecs =
calculateTotalSpecsForArray(tempVehicles);
        numberOfAxles += sign*tempSpecs[1];
        numberOfPassengers += sign*tempSpecs[2];
        numberOfVehicles += sign*tempSpecs[3];
        weight += sign*tempSpecs[0];
    }
    private boolean doIDsMatch(Vehicle[] existingVehicles, Vehicle[]
newVehicles) {
```

```
        for (Vehicle vehicle: existingVehicles) {
            for (Vehicle newVehicle: newVehicles) {
                if (vehicle.getUniqueID() == newVehicle.getUniqueID())
{
                    return true;
                }
            }
        }
        return false;
    }
     */
    public boolean changePassengersOnVehicle(Vehicle vehicle, int
newNumberOfPassengers) {
        if ((newNumberOfPassengers-
vehicle.getNumberOfPassengers())+newNumberOfPassengers >
maximumNumberOfPassengers) {
            return false;
        }
        else {
            vehicle.setNumberOfPassengers(newNumberOfPassengers);
            numberOfPassengers += newNumberOfPassengers -
vehicle.getNumberOfPassengers();
            return true;
        }
    }
    public String weightReport() {
        return ("The weight left on the ferry is " + (maxWeight -
weight) + " Kg");
    }
    public String vehicleList() {
        String list = "";
        DecimalFormat df = new DecimalFormat("00,000.00");
```

```
        for(Vehicle vehicle: vehicles) {
            String patternedWeight = df.format(vehicle.getWeight());
            list += vehicle.getUniqueID() + "\t" + patternedWeight + "
kg\n";
        }
        list+= "Total Weight:\t" + weight + "\n";
        list+= "Total Axles:\t" + numberOfAxles + "\n";
        list+= "Total Passengers:\t" + numberOfPassengers+ "\n";
        return list;
    }
}
```

Section C

Source Code:

```
/**
 This class is the driver the program.
 @author Isaac Shoebottom (3429069)
 */
public class Driver {


    public static void main(String[] args) {
        Vehicle vehicle1 = new Vehicle(2, 4, 450.45);
        Vehicle vehicle2 = new Vehicle(2, 8, 636.30);
        Vehicle vehicle3 = new Vehicle(4, 12, 4500);
        Vehicle vehicle4 = new Vehicle(16, 4, 100);
        Vehicle vehicle5 = new Vehicle(16, 40, 100);
        Ferry ferry = new Ferry(3, 10, 16, 3333);



        if (ferry.addVehicle(vehicle1)) {
            System.out.println("Vehicle " + vehicle1.getUniqueID() + "
added");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle1.getUniqueID());
        }
        if (ferry.addVehicle(vehicle1)) {
            System.out.println("Vehicle " + vehicle1.getUniqueID() + "
added!");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle1.getUniqueID());
        }
```

```
        if (ferry.addVehicle(vehicle2)) {
            System.out.println("Vehicle " + vehicle2.getUniqueID() + "
added!");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle2.getUniqueID());
        }
        if (ferry.addVehicle(vehicle3)) {
            System.out.println("Vehicle " + vehicle3.getUniqueID() + "
added!");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle3.getUniqueID());
        }
        if (ferry.addVehicle(vehicle3)) {
            System.out.println("Vehicle " + vehicle3.getUniqueID() + "
added!");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle3.getUniqueID());
        }
        if (ferry.addVehicle(vehicle4)) {
            System.out.println("Vehicle " + vehicle4.getUniqueID() + "
added!");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle4.getUniqueID());
        }
        if (ferry.addVehicle(vehicle5)) {
            System.out.println("Vehicle " + vehicle5.getUniqueID() + "
added!");
        } else {
            System.out.println("Couldn't add Vehicle " +
vehicle5.getUniqueID());
```

```
        }
        if (ferry.removeVehicle(vehicle5)) {
            System.out.println("Vehicle " + vehicle5.getUniqueID() + "
removed!");
        } else {
            System.out.println("Couldn't remove Vehicle " +
vehicle5.getUniqueID());
        }
        if (ferry.removeVehicle(vehicle2)) {
            System.out.println("Vehicle " + vehicle2.getUniqueID() + "
removed!");
        } else {
            System.out.println("Couldn't remove Vehicle " +
vehicle2.getUniqueID());
        }


        if (ferry.changePassengersOnVehicle(vehicle1,7)) {
            System.out.println("Vehicle " + vehicle1.getUniqueID() + "
passenger's changed!");
        } else {
            System.out.println("Couldn't change passengers");
        }
        if (ferry.changePassengersOnVehicle(vehicle1, 69)) {
            System.out.println("Vehicle " + vehicle1.getUniqueID() + "
passenger's changed!");
        } else {
            System.out.println("Couldn't change passengers");
        }
        if (ferry.changePassengersOnVehicle(vehicle4, 30)) {
            System.out.println("Vehicle " + vehicle5.getUniqueID() + "
passenger's changed!");
        } else {
            System.out.println("Couldn't change passengers");
```

```
        }


        System.out.println(ferry.weightReport());
        System.out.println(ferry.vehicleList());


    }

}
```

Section D

Sample Output:

```
Vehicle 101 added

Couldn't add Vehicle 101

Vehicle 102 added!

Couldn't add Vehicle 103

Couldn't add Vehicle 103

Couldn't add Vehicle 104

Couldn't add Vehicle 105

Couldn't remove Vehicle 105

Vehicle 102 removed!

Vehicle 101 passenger's changed!

Couldn't change passengers

Couldn't change passengers

The weight left on the ferry is 2882.55 Kg

101   00,450.45 kg

Total Weight:    450.45000000000005

Total Axles:     2

Total Passengers:     2
```

I know there is still bugs in my code, and no Javadoc as well but I had work the past few days and this is about as good as I can get it in the time I have left to do it, I hope the final result shows my understanding of the subject even if there are some bugs. There is some commented out code that I wanted to experiment with but ultimately didn't have enough time or didn't know if it was allowed in the assignment.