CT2106 OOP Assignment 1

30/09/22

The Code:

TestCar Class

```
/**
 * This class is used to generate output for the simulation
 *
 * @author Tim
 * @version V1.0
 */
public class TestCar
{
   public static void main(String[] args)
   {
      Car car = new Car("X7");
      Engine engine = new Engine("DR9", 43);
      car.add(engine);
      Wheel wheel = new Wheel("Wichelin15", 15);
      car.add(wheel);
      car.setFuel(100);
      System.out.printf("Current fuel: %.2f\n",car.getFuel());
      car.drive();
      car.printState();
      car.setFuel(50);
      System.out.printf("Current fuel: %.2f\n",car.getFuel());
      car.drive();
      car.printState();
   }
}
```

Car Class

```
/**
 * Car class for the car simulation.
 *
 * @author Tim
 * @version V1.0
 */
public class Car
{
   private String carName;
   private double distance;
   private double totalDistance;

   //A car has an engine and can access it
   private Engine engine;



   /**
    * Constructor
    */
   public Car(String carName)
   {
      this.carName = carName;
      //Setting total distance to 0 as the car hasnt moved after initial construction
      totalDistance = 0;
   }

   /**
      Car Methods
    */
```

```
  public void drive() {
     //At the beginning of each drive, new fuel is added and the distance is reset
    distance = 0;
    //The car calls on the engine to begin combustion and start driving.
    distance = engine.call();
    //The engine uses all of the fuel availible
    engine.setFuel(0);

    totalDistance += distance;
  }


  public void printState() {
    System.out.println("Configuration: Car Body "+carName);
    System.out.println("Engine name: "+engine.getEngineName());
    System.out.printf("Engine turns per litre: %.2f \n",engine.getEngineTpl());
    System.out.printf("Engine's total turn count: %d \n",engine.getEngineTurns());
    System.out.println("Wheel name: "+engine.getWheelName());
    System.out.printf("Wheel radius: %.2f\n",engine.getWheelRadius());
    System.out.printf("Wheel circumference (distance per turn): %.2f\n",engine.getWheelCir());
    System.out.printf("Distance this trip: %.2f\n",distance);
    System.out.printf("Total distance Travelled: %.2f\n",totalDistance);
    System.out.printf("Current fuel Status: %.2f\n",engine.getFuel());
    System.out.println();
  }

/**
    Car Accessor and Mutator Methods
  */
```

```
 //Add method to add specific engine object to the car
    public void add(Engine engine){
        this.engine = engine;
    }

    //Another add method, but with a wheel argument.Since only the engine class has a wheel object, it will send the wheel as a parameter
    //to the engine's add method.
     public void add(Wheel wheel){
        engine.add(wheel);
    }

    //Methods used to retrieve the current fuel levels from the engine class by calling the engines mutator/accessor methods
    public void setFuel(double fuelLevel) {
        engine.setFuel(fuelLevel);
    }

    public double getFuel(){
        return engine.getFuel();
    }
    }
```

Engine Class

```
/**
 * Engine class for the car simulation.
 *
 * @author Tim
 * @version V1.3
 */
public class Engine
{
   private int totalTurns;
   private double tpl;
   private String name;

    //I've decided to put the fuel level inside the engine class as fuel is a fundamental part of the combustion process in a car. Ive also found it to design this simulation with
    //the fuel inside the engine object.
   private double fuelLevel;

   //An engine has a wheel.
   private Wheel wheel;

   /**
    * Constructor
    */
   public Engine(String name,double tpl)
   {
      this.tpl = tpl;
      this.name = name;
   }
```

```
/**
    * Methods
    */

   public double call(){
      //As totalTurns needs to be incremented on each turn() method call, i've used a for loop to turn the wheels and keep track of the number of total turns.
      //The wheel will be turned as many times as the tpl specifies by the amount of total fuel available.
      for(int i = 0; i<(fuelLevel*tpl); i++){
         wheel.turn();
         totalTurns++;
      }
     //Once the engine has run out of fuel, the total distance travelled is returned to the car for it to report to the driver.
      return fuelLevel*(wheel.turn()*tpl);
   }

   /**
      Engine Accessor and Mutator Methods
    */

   public void setFuel(double fuelLevel) {
      this.fuelLevel = fuelLevel;
   }

   public double getFuel(){
      return fuelLevel;
   }

   public String getEngineName(){
      return name;
   }
```

```
    public double getEngineTpl(){
        return tpl;
    }

    public int getEngineTurns(){
        return totalTurns;
    }

    public void add(Wheel wheel){
        this.wheel = wheel;
    }



    /**
     * Wheel Accessor Methods
     */

    //Methods used to retrieve specifc details from the wheel by calling the wheel class' mutator/accessor
methods since only the engine can access the wheel
    public String getWheelName(){
        return wheel.getName();
    }

     public double getWheelRadius(){
        return wheel.getRadius();
    }

    public double getWheelCir(){
        return wheel.turn();
    }

}
```

Wheel Class

```
/**
 * Wheel class for the car simulation.
 *
 * @author Tim
 * @version V1.0
 */
public class Wheel
{
   private double radius;
   private String name;
   private double circumference;

   /**
    * Constructor for objects of class Wheel
    */
   public Wheel(String name, double radius)
   {
      this.radius = radius;
      this.name = name;
      //Upon the construction of the wheel, the circumference is calculated.
      circumference = 2 * Math.PI * radius;
   }


    /**
     Accessor Methods
    */
```

```
  //These methods are used to report details back to the engine class.
    public double turn() {
        return circumference;
    }

    public String getName(){
      return name;
    }

    public double getRadius(){
      return radius;
    }

}
```

Code Outline:

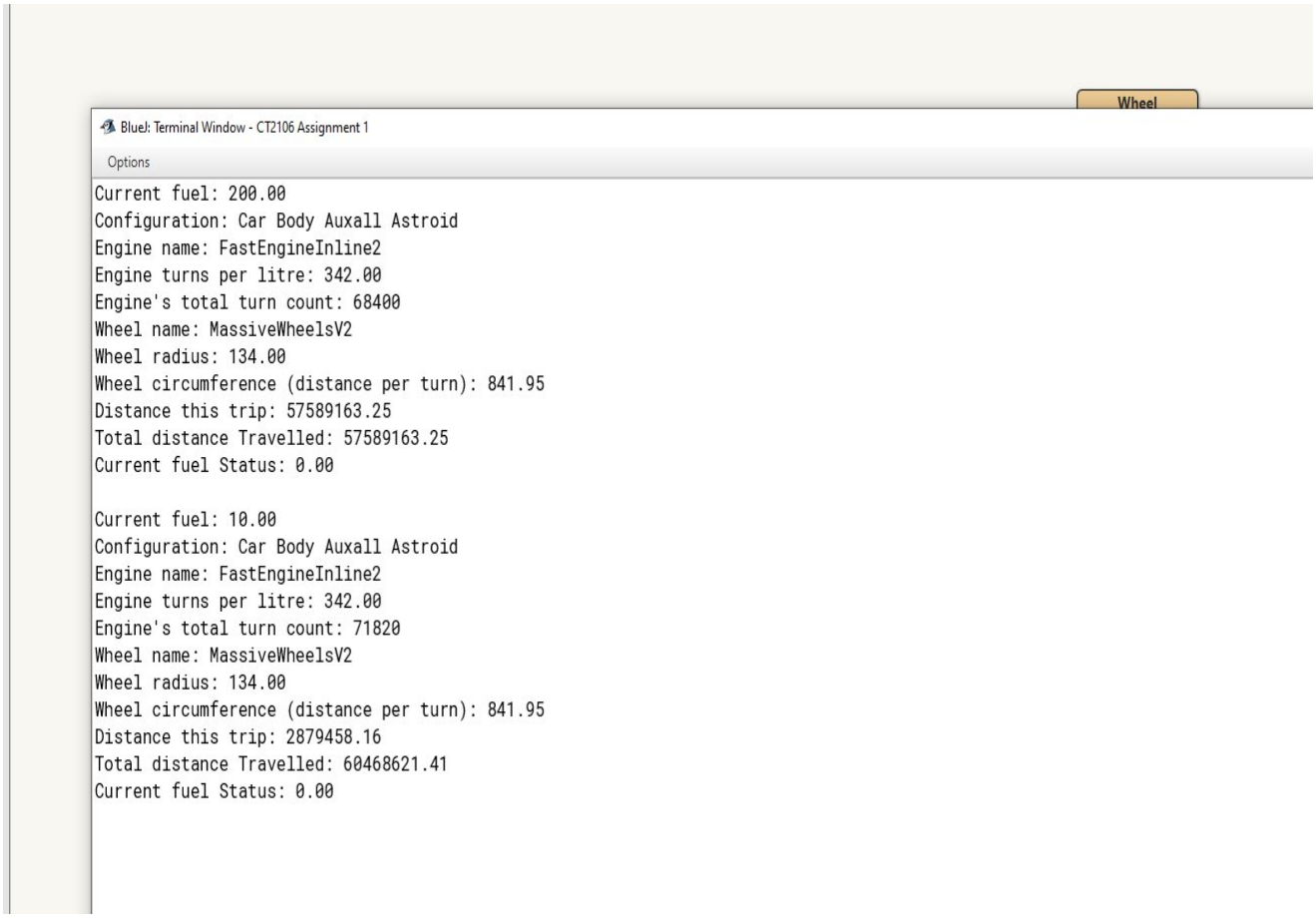
The code revolves around three main classes, the car, the engine, and the wheel. The composition relationship between them are as follows, the car **has an** engine, and the engine **has a** wheel. Instance variables are created in the respective classes that refer to the other class that belongs to it. For example, the wheel could not exist without the engine.

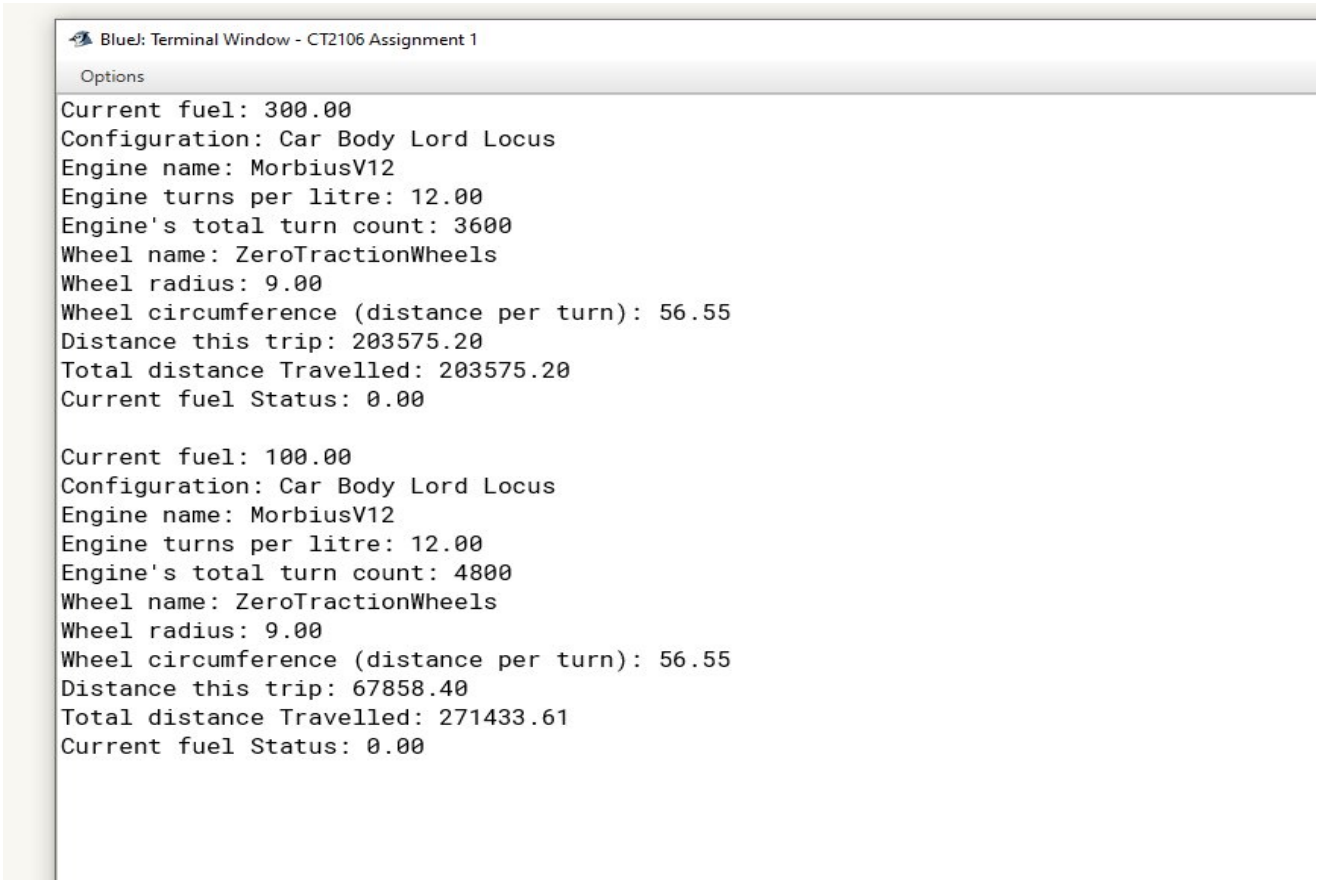
The car class possesses two add methods that are used to add the engine object to the car and the wheel object to the engine. Both objects were created, and their parameters were set in main (TestCar). The car class also has a mutator and accessor method for fuel. The fuel level variable belongs to the engine class, so through these methods, the car class accesses the engine class and calls the engine methods to retrieve or set the fuel level.

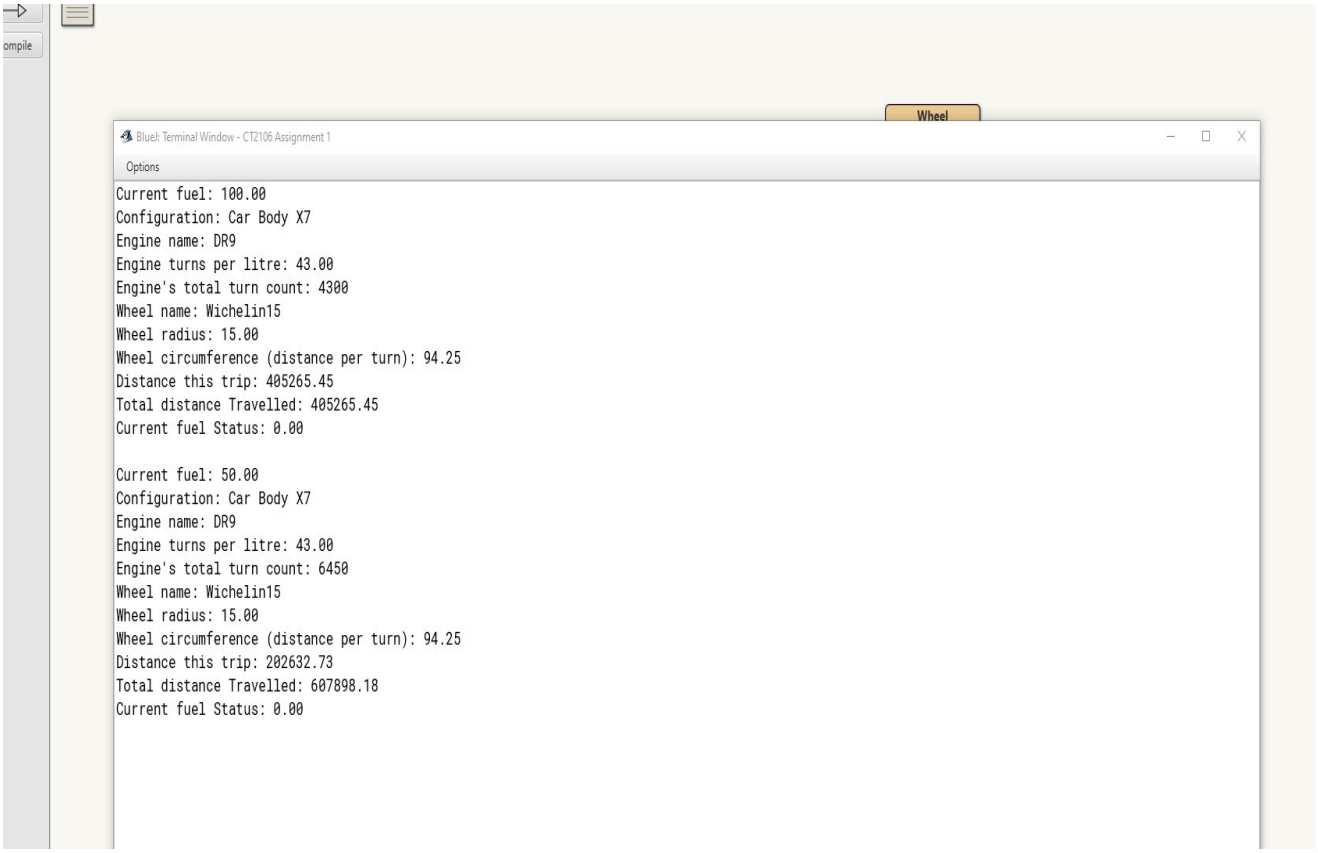
The drive method in the car class calls upon the engine to start driving (i.e., start combustion and turn the wheels) using the call method inside the engine class. Using the TPL of the engine and the fuel level to calculate the max number of turns, the engine turns the wheels and increments the total turns variable as it loops until it runs out of fuel. Upon using all the fuel, the call method returns the distance travelled in that session. The turn method inside the wheel class returns the distance travelled in one revolution of the wheel.

The print state method in the car class prints out the status of the car to the terminal. Since the car class can't retrieve the wheel variable values, it must get them through the engine class. To retrieve the values for the wheel variables from the car class, the method calls upon another method in the engine class, which calls another method in the wheel class to return specific wheel values back up the chain to be printed in the car class.

Output Screenshots:






Wheel
BlueJ: Terminal Window - CT2106 Assignment 1
Options
Current fuel: 100.00
Configuration: Car Body X7
Engine name: DR9
Engine turns per litre: 43.00
Engine's total turn count: 4300
Wheel name: Wichelin15
Wheel radius: 15.00
Wheel circumference (distance per turn): 94.25
Distance this trip: 405265.45
Total distance Travelled: 405265.45
Current fuel Status: 0.00
Current fuel: 50.00
Configuration: Car Body X7
Engine name: DR9
Engine turns per litre: 43.00
Engine's total turn count: 6450
Wheel name: Wichelin15
Wheel radius: 15.00
Wheel circumference (distance per turn): 94.25
Distance this trip: 202632.73
Total distance Travelled: 607898.18
Current fuel Status: 0.00