R·I·T

Rochester Institute of Technology
Golisano College of Computing and Information Sciences
Department of Information Sciences & Technology

ISTE-200 Java for Programmers

File I/O

Homework Assignment # 6

Due next week class 7a at beginning of class

Objectives:

This homework will have you gain experience with File I/O, using code from others, and maintaining your previous homework code. It will be heavily graded on the File I/O, Java docs, the appearance of the code and outputs. Make sure you have all the requirements completed and correct. If you have any questions, please ask or suffer the consequences.

Maintenance of code:

This homework builds on the existing code you wrote before. Do not rewrite that code – but some of the code may need to be updated. Any errors indicated in previous homework(s), should be corrected as they may affect your ability to complete this homework. Maintenance of code consists of fixing and enhancing existing code. ***Should you be falling behind in the homework, contact the instructor to setup extra office hours or lab time to help you through the problems with the homework.***

Enhancements:

Starting with your Ordering System classes the enhancements are:

1) When your program starts have it look for a file, VehicleOrders.dat. If the file exists, open the file and read the orders into the array list.
2) When the user is finished with the program and wants to exit, the program writes a new file with the same name. This way the list of orders does not go away and the program ends. Delete the file if you want to start over again.
   Also write a text file report VehicleOrders.txt which looks like the summary output written by the toString methods at the end of the program.
3) The subclasses should have a constant, stating what they are. This is passed to the Vehicle class as shown in this code example from Boat:

```
final static String VEHICLE_IS = "Boat";
public Boat()
{
     super( VEHICLE_IS );
     // rest of constructor code...
```

4) Vehicle's .toString() method should look similar to this code. This shows the use of getting the VEHICLE_IS value, and the mpgEstimate(). This code is in Vehicle, but calls the mpgEstimate() of the subclasse's type, such as Boat, Car, Truck.

```
/** returns the Mode, Color, Cost (formatted) and MPG/GPH (formatted) */
```

```
public String toString()
{
   return                   getVehIs() +
         "\n\tModel:   " + getModel() +
         "\n\tColor:   " + getColor() +
         "\n\tCost:    " + String.format( "$%,.2f", getCost() ) +
         "\n\tMPG/GPH: " + String.format( "%,.1f", mpgEstimate() );

}
```

The subclass toString should look similar to this Boat example:

```
/** returns information about this Boat */
public String toString()
{
   return super.toString() +     // Super supplies model, color, cost
        "\n\tType:    "+  getBoatType() +
        "\n\tMade of: "+  getBoatMadeOf();

} // end toString
```

Requirements summary:

- Read a file of order data written from the previous run of your program, if it exists.
- Upon exit of the program, write the ordered data to a file that can be read in upon the next run of this program.
- All of previous HW requirements, fixed.
- Java Docs – modify where needed, and add Java Docs to the new code. Start to follow the "JavaDoc tag usage".
- Program names: Orders.java, Truck.java, Car.java, Boat.java, Vinfo.java, your new vehicle, and another student's vehicle class.
- The main program is **Orders**, not Order!
- There should be no duplication of code in any one class, and little duplication of code between classes. Should be using inheritance and common methods wherever possible.
- Comments – Make comments meaningful. Saying that "Number for stupid equation" is not considered an appropriate, nor meaningful comment.
- Use String.format for formatting text for output only. There is no need to format the input before storing the number back into (itself) the double attribute.

Submitting your work:

- Submit all of your java and class files to the homework DropBox.
- Do **not** include your JavaDoc files to the DropBox.

Sample Output: (Bolded text was entered by the user)

Your execution will include Truck, Car, Boat, your new class, and one other taken from the Q&A conference.

```
C:\>java Orders
Your Name
218-01 Q:024
Homework # 6
Prof. Floeser

Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?c               ← Enter a valid car – easy!
Entering Car order:
      Model:  Ford
      Color:  Green
      Cost:   14000
What type of car is this?
      1. Sedan
      2. Coupe
      3. Wagon
      Choice:     2
Does this car have a towing package?
      1. Towing package
      2. No towing package
      Choice:     2                                    ← Valid car entry done

Do you want to order another vehicle? y
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?S               ← Unknown type - Ignored

Invalid vehicle type.
Try again.
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?Scooter  ← All other input ignored, model

Invalid vehicle type.
Try again.
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?Pink     ← All other input ignored, color

Invalid vehicle type.
Try again.
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?3000     ← All other input ignored, cost

Invalid vehicle type.
Try again.
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?2        ← All other input ignored, choice 1

Invalid vehicle type.
Try again.
```

```
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?1          ← All other input ignored, choice 2

Invalid vehicle type.
Try again.
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?y          ← All other input ignored, yes more

Invalid vehicle type.
Try again.
Do you want to order a Truck, Car, Boat, Zamboni, Motorcycle, or Lawnmower
       Enter one of: T, C, B, Z, M, L ?T          ← Back on track, valid truck
Entering Truck order:
      Model:  Dodge
      Color:  Red
      Cost:   $25,234.56                          ← invalid Cost, try again
Invalid cost, enter numbers, no $ or commas.
      Cost:   25234.56                            ← valid Cost, all good
What size truck is this?
      1. Half-ton
      2. Full ton
      Choice:     9                               ← Invalid choice
What size truck is this?
      1. Half-ton
      2. Full ton
      Choice:     -4                              ← Invalid choice
What size truck is this?
      1. Half-ton
      2. Full ton
      Choice:     yes                             ← Invalid choice, words caught
Need a counting number from 1 through 2 please try again.
What size truck is this?
      1. Half-ton
      2. Full ton
      Choice:     2                               ← Valid truck size
What is the engine size of the truck?
      1. Really big
      2. Not so big
      Choice:     99                              ← Invalid choice
What is the engine size of the truck?
      1. Really big
      2. Not so big
      Choice:     -44                             ← Invalid choice
What is the engine size of the truck?
      1. Really big
      2. Not so big
      Choice:     no                              ← Invalid choice, words caught
Need a counting number from 1 through 2 please try again.
What is the engine size of the truck?
      1. Really big
      2. Not so big
      Choice:     1                               ← Valid choice, done truck entry
Do you want to order another vehicle? N           ← Done, print to screen and file
```

```
Car
      Model:   Ford
      Color:   Green
      Cost:    $14,000.00
      MPG/GPH: 25.2
      Type:    Coupe
      Towing:  No towing package

Truck
      Model:   Dodge
      Color:   Red
      Cost:    $25,234.56
      MPG/GPH: 34.1
      Load:    Full ton
      Engine:  Really big


Thank you for using Your Name's Ordering System.

C:\>
```

After the above run, the contents of file **VehicleOrder.txt** is:

```
Car
      Model:   Ford
      Color:   Green
      Cost:    $14,000.00
      MPG/GPH: 25.2
      Type:    Coupe
      Towing:  No towing package

Truck
      Model:   Dodge
      Color:   Red
      Cost:    $25,234.56
      MPG/GPH: 34.1
      Load:    Full ton
      Engine:  Really big
```

ISTE-200 Homework 6 – File I/O grade sheet

Item	Points available	Points earned
Order.java		
Read orders from file, load data into ArrayList, for new orders to be added to it.	20	
Upon exit of program write a file, VehicleOrders.**dat**, that can be read on next run and loaded into the ArrayList of vehicles	20	
Upon exit of program, write a file, VehicleOrders.**txt**, is a report file of the vehicles ordered. This looks like the summary report printed.	10	
Javadocs and @ tags properly used	5	
Vinfo.java		
Javadocs and @ tags properly used	5	
Vehicle.java		
Subclasses should have a constant, stating what they are. This is passed to the Vehicle class	5	
Javadocs and @ tags properly used	5	
Your new class		
Subclasses should have a constant, stating what they are. This is passed to the Vehicle class	5	
Javadocs and @ tags properly used	5	
Other one class		
Subclasses should have a constant, stating what they are. This is passed to the Vehicle class	5	
Javadocs and @ tags properly used	5	
Boat.java, Truck.java, Car.java		
Subclasses should have a constant, stating what they are. This is passed to the Vehicle class	5	
Javadocs and @ tags properly used	5	
Deductions: Violations of coding standards or poor documentation		
Total	100	