

## PRACTICAL (QUESTION #6)

### UseVehicle.java test documentation

---

Submitted by:       Datoon, Philip Bryan B.  
                          131311399

---

The following shows a table of test results for UseVehicle.java, specifying possible inputs and expected results for each function of the program.

Function	Input	Expected result(s)
1. User is asked to enter value for <i>vehicle</i> 's number of wheels.	wheels = eight	Value is invalid. Program must ask user to enter data again for vehicle's number.
	wheels = 8.0	
	wheels = 8d	
	wheels = 0	Value is zero and is invalid. Program must ask user to enter data again for vehicle's average miles per gallon.
	wheels = -8	Value is negative and is invalid. Program must ask user to enter data again for vehicle's average miles per gallon.
	wheels = 8	Input is valid, and must be accepted by the program.
2. User is asked to enter value for <i>vehicle</i> 's average miles per gallon	miles = one hundred	Value is invalid. Program must ask user to enter data again for vehicle's average miles per gallon.
	miles = 100.0	
	miles = 100x	
	miles = 0	Value is zero and is invalid. Program must ask user to enter data again for vehicle's average miles per gallon.
	miles = -100	Value is negative and is invalid. Program must ask user to enter data again for vehicle's average miles per gallon.
	miles = 100	Input is valid, and must be accepted by the program.
3. User is asked to enter value for <i>car</i> 's average miles per gallon.	All input for average miles per gallon used to call getMiles() method. All possible input is the same as that of function #2.	Expected results must be the same as that of function #2.

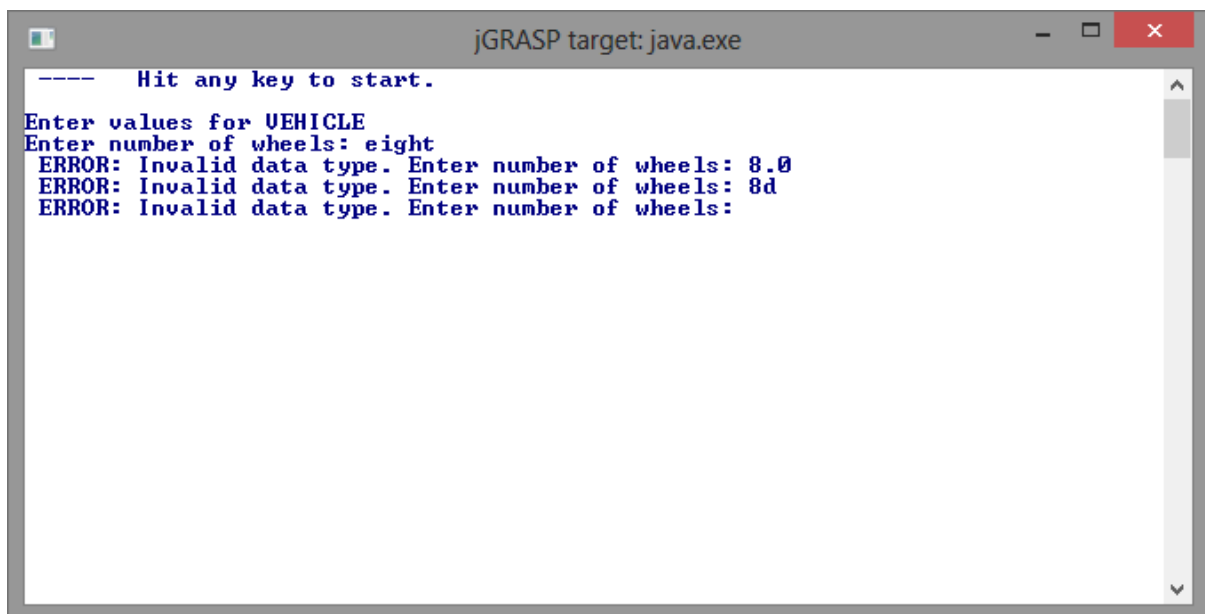
4. User is asked to enter value for <i>motorcycle</i> 's average miles per gallon.	All input for average miles per gallon used to call <code>getMiles()</code> method. All possible input is the same as that of function #2.	Expected results must be the same as that of function #2.
5. Display the results.	N/A	Once all the inputs are valid, results and values of the input must be displayed on the screen.

## ACTUAL RESULTS

The following section shows images or screenshots of the actual results of the program for functions specified in the table in previous page.

### Input values for `Vehicle` object

`Vehicle` class requires two values: number of wheels, and average miles per gallon. Program starts by asking user to enter number of wheels for the vehicle object.

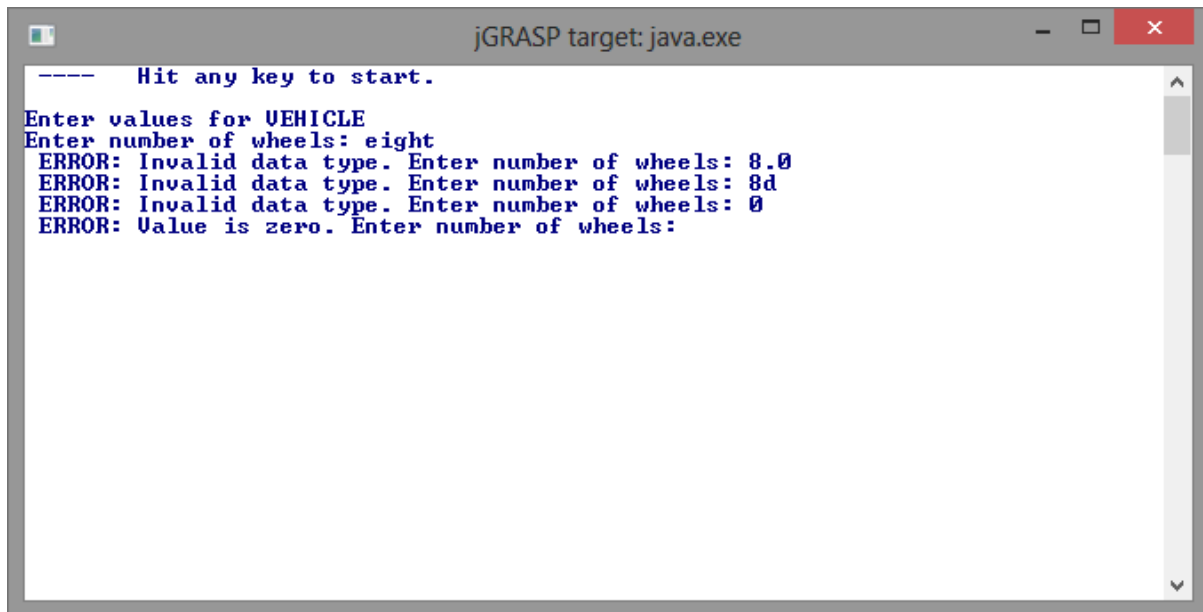


```

jGRASP target: java.exe
----- Hit any key to start.
Enter values for VEHICLE
Enter number of wheels: eight
ERROR: Invalid data type. Enter number of wheels: 8.0
ERROR: Invalid data type. Enter number of wheels: 8d
ERROR: Invalid data type. Enter number of wheels:

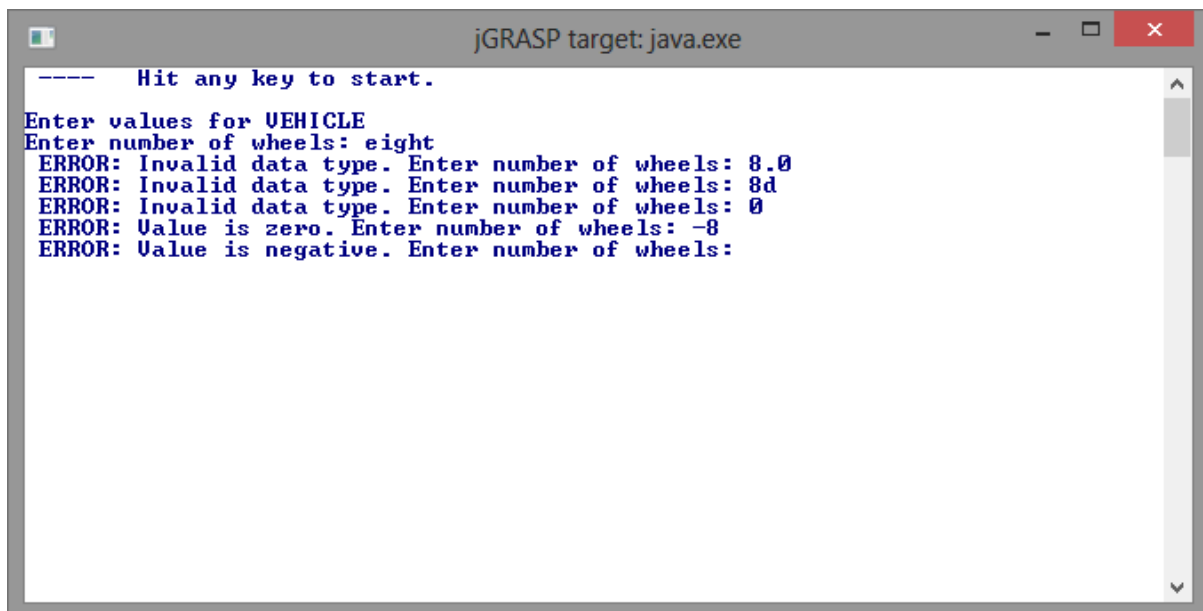
```

If the user attempted to enter String, double, or any value with symbols and characters, an error message should appear and the program should ask the user again to enter new and valid data.



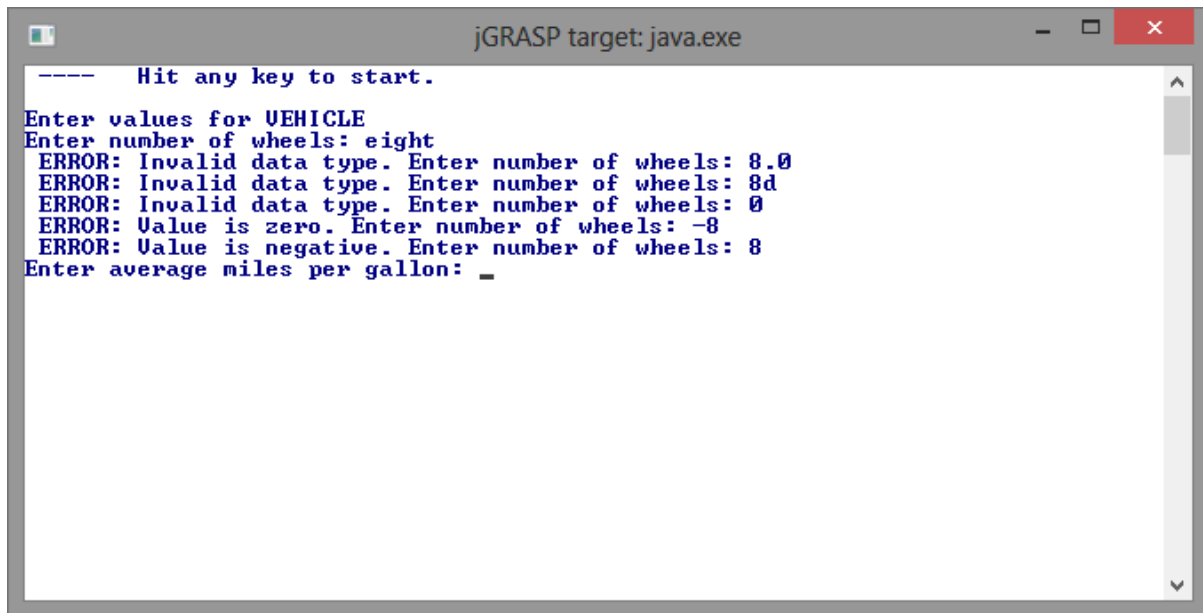
```
----- Hit any key to start.
Enter values for VEHICLE
Enter number of wheels: eight
ERROR: Invalid data type. Enter number of wheels: 8.0
ERROR: Invalid data type. Enter number of wheels: 8d
ERROR: Invalid data type. Enter number of wheels: 0
ERROR: Value is zero. Enter number of wheels:
```

If user attempted to enter zero, program accepts the integer value. However, zero should not be accepted as a value for number of wheels, and is still invalid. Thus, another error message should appear and the program should ask the user again to enter new and valid data.



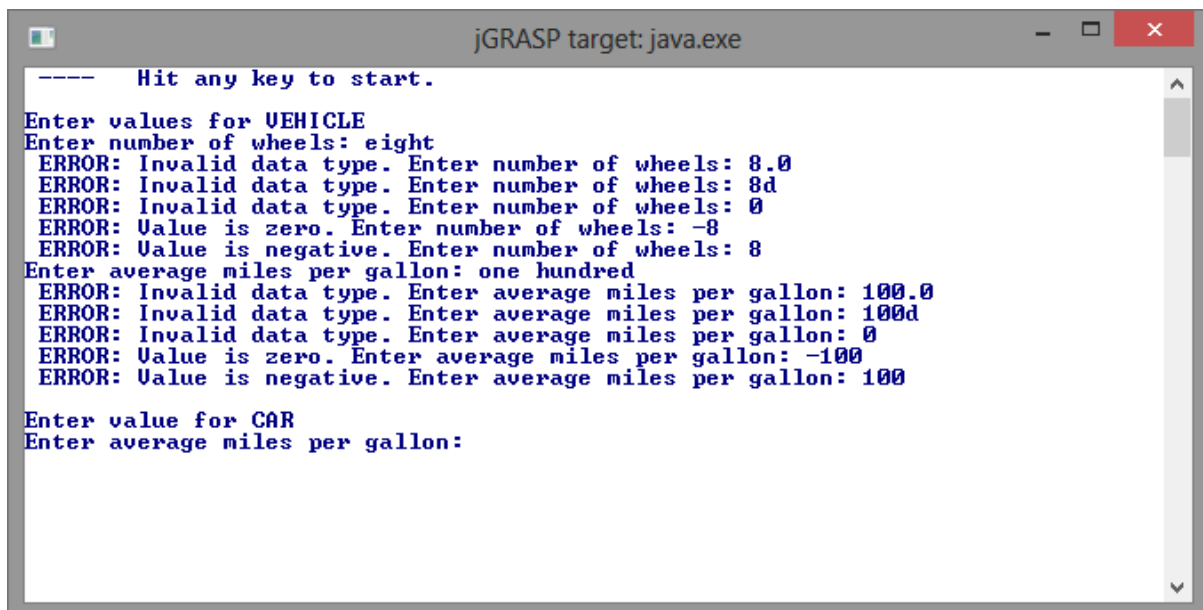
```
----- Hit any key to start.
Enter values for VEHICLE
Enter number of wheels: eight
ERROR: Invalid data type. Enter number of wheels: 8.0
ERROR: Invalid data type. Enter number of wheels: 8d
ERROR: Invalid data type. Enter number of wheels: 0
ERROR: Value is zero. Enter number of wheels: -8
ERROR: Value is negative. Enter number of wheels:
```

If user attempted to enter a negative value, program accepts the integer value. However, any negative integer should not be accepted as a value for number of wheels, and is still invalid. Thus, an error message should be displayed and the program should ask the user again to enter new and valid data.



```
---- Hit any key to start.
Enter values for VEHICLE
Enter number of wheels: eight
ERROR: Invalid data type. Enter number of wheels: 8.0
ERROR: Invalid data type. Enter number of wheels: 8d
ERROR: Invalid data type. Enter number of wheels: 0
ERROR: Value is zero. Enter number of wheels: -8
ERROR: Value is negative. Enter number of wheels: 8
Enter average miles per gallon: _
```

Once the user entered a valid input, program accepts the value and then asks user to input average miles per gallon for vehicle object.

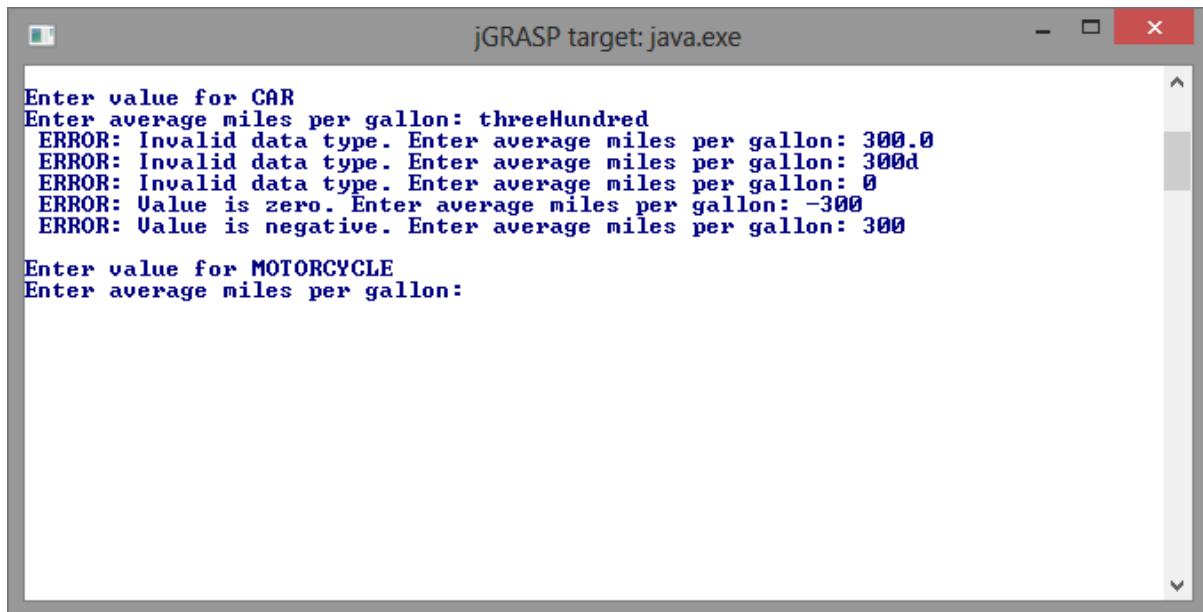


```
---- Hit any key to start.
Enter values for VEHICLE
Enter number of wheels: eight
ERROR: Invalid data type. Enter number of wheels: 8.0
ERROR: Invalid data type. Enter number of wheels: 8d
ERROR: Invalid data type. Enter number of wheels: 0
ERROR: Value is zero. Enter number of wheels: -8
ERROR: Value is negative. Enter number of wheels: 8
Enter average miles per gallon: one hundred
ERROR: Invalid data type. Enter average miles per gallon: 100.0
ERROR: Invalid data type. Enter average miles per gallon: 100d
ERROR: Invalid data type. Enter average miles per gallon: 0
ERROR: Value is zero. Enter average miles per gallon: -100
ERROR: Value is negative. Enter average miles per gallon: 100
Enter value for CAR
Enter average miles per gallon:
```

Error handling and checking the validity of input for average miles per gallon works the same as to that of for number of wheels. Once validated, program will then ask user to enter average miles per gallon for Car object.

## Input values for Car object

Car class requires only one value: average miles per gallon.



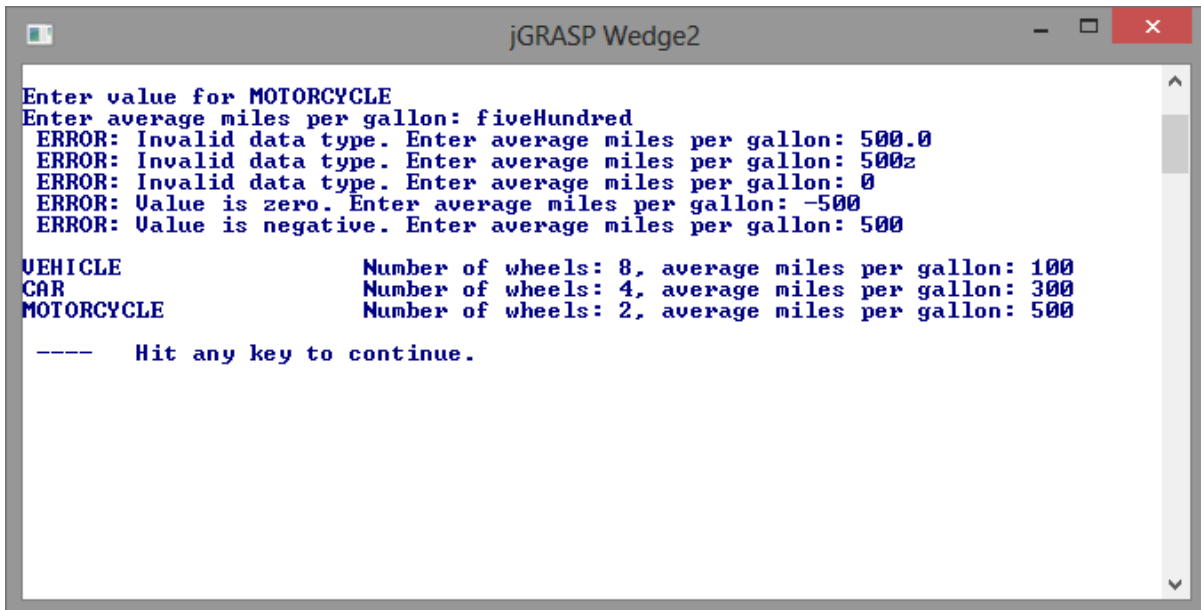
```
Enter value for CAR
Enter average miles per gallon: threeHundred
ERROR: Invalid data type. Enter average miles per gallon: 300.0
ERROR: Invalid data type. Enter average miles per gallon: 300d
ERROR: Invalid data type. Enter average miles per gallon: 0
ERROR: Value is zero. Enter average miles per gallon: -300
ERROR: Value is negative. Enter average miles per gallon: 300

Enter value for MOTORCYCLE
Enter average miles per gallon:
```

Error handling and checking the validity of input for average miles per gallon works the same as to that of for number of wheels and average miles per gallon in `Vehicle` object. Once validated, program will then ask user to enter average miles per gallon for `MotorCycle` object.

## Input values for Motorcycle object

MotorCycle class requires only one value: average miles per gallon.



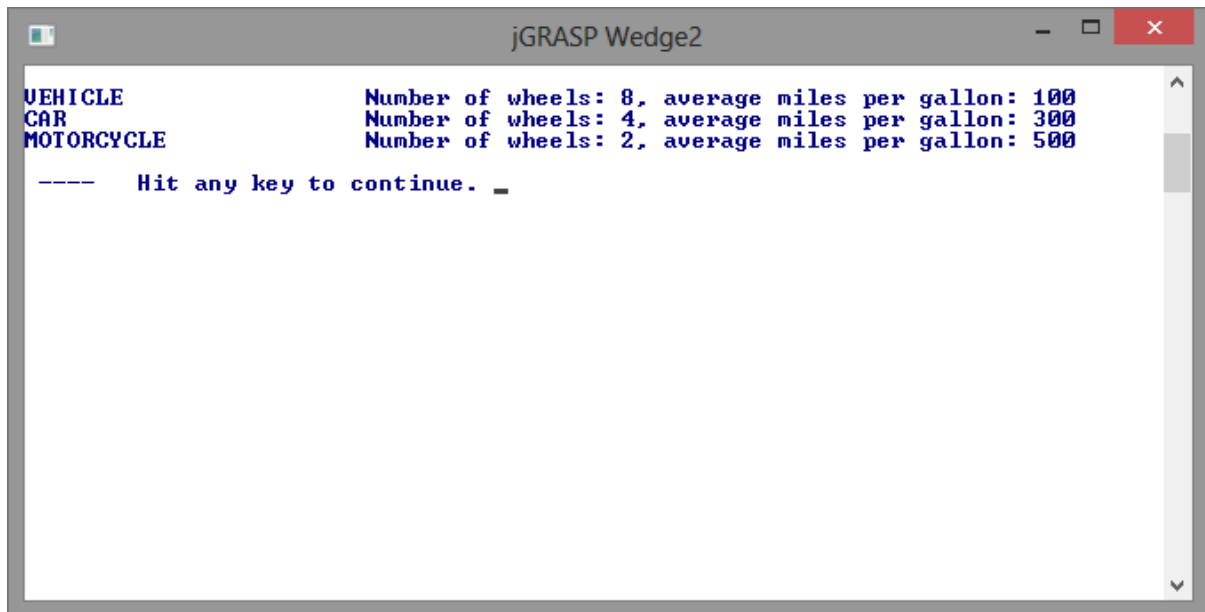
```
Enter value for MOTORCYCLE
Enter average miles per gallon: fiveHundred
ERROR: Invalid data type. Enter average miles per gallon: 500.0
ERROR: Invalid data type. Enter average miles per gallon: 500z
ERROR: Invalid data type. Enter average miles per gallon: 0
ERROR: Value is zero. Enter average miles per gallon: -500
ERROR: Value is negative. Enter average miles per gallon: 500

VEHICLE          Number of wheels: 8, average miles per gallon: 100
CAR              Number of wheels: 4, average miles per gallon: 300
MOTORCYCLE       Number of wheels: 2, average miles per gallon: 500

---- Hit any key to continue.
```

Error handling and checking the validity of input for average miles per gallon works the same as to that of for number of wheels and average miles per gallon in Vehicle object. Once validated, program will then ask user to enter average miles per gallon for Motorcycle object.

## Displaying inputs and results

A screenshot of a Java Swing window titled "jGRASP Wedge2". The window contains a text area with the following text:

```
VEHICLE          Number of wheels: 8, average miles per gallon: 100
CAR              Number of wheels: 4, average miles per gallon: 300
MOTORCYCLE       Number of wheels: 2, average miles per gallon: 500

---- Hit any key to continue. _
```

The text is displayed in a monospaced font. The window has standard OS controls (minimize, maximize, close) in the title bar.

Once all the inputs are valid, results and values of the input must be displayed on the screen. The program is expected that the number of wheels for `Car` and `MotorCycle` objects should be 4 and 2, respectively, as default values set through their own class.