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 ***vehicle***'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 ***vehicle***'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 ***car***'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.

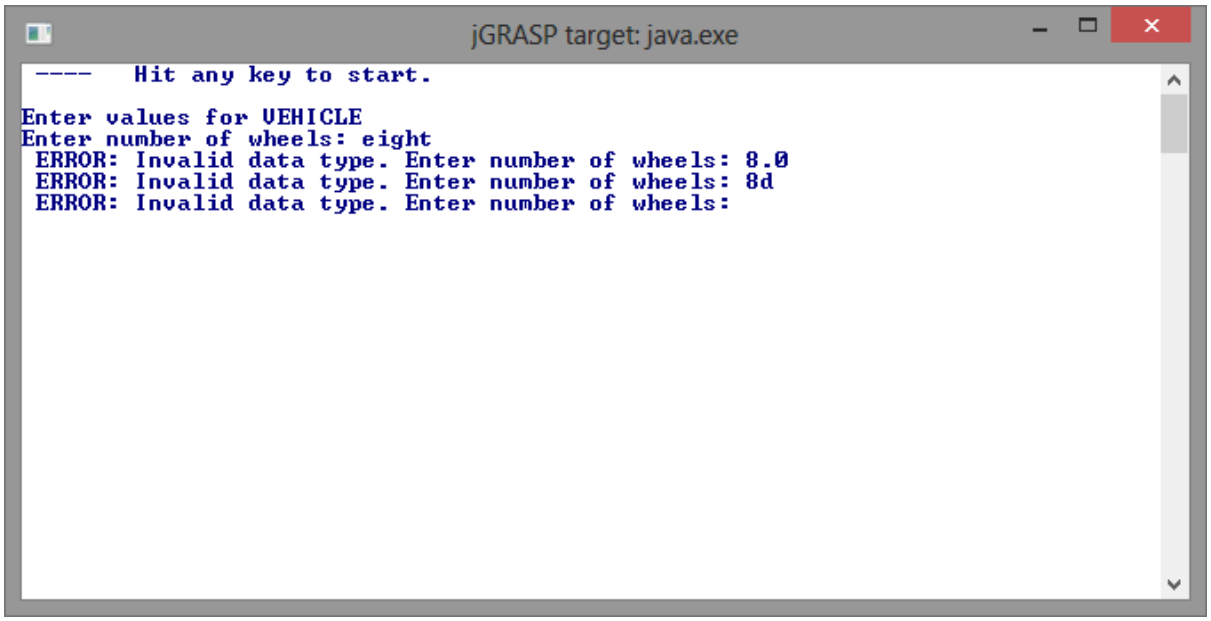
4. User is asked to enter value for ***motorcycle***'s average miles per gallon.	All input for average miles per gallon used to call `getMiles()` method. All possible input is the same as that of function #2.	Expected results must be the same as that of function #2.
5. Display the results.	`N/A`	Once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

ACTUAL RESULTS

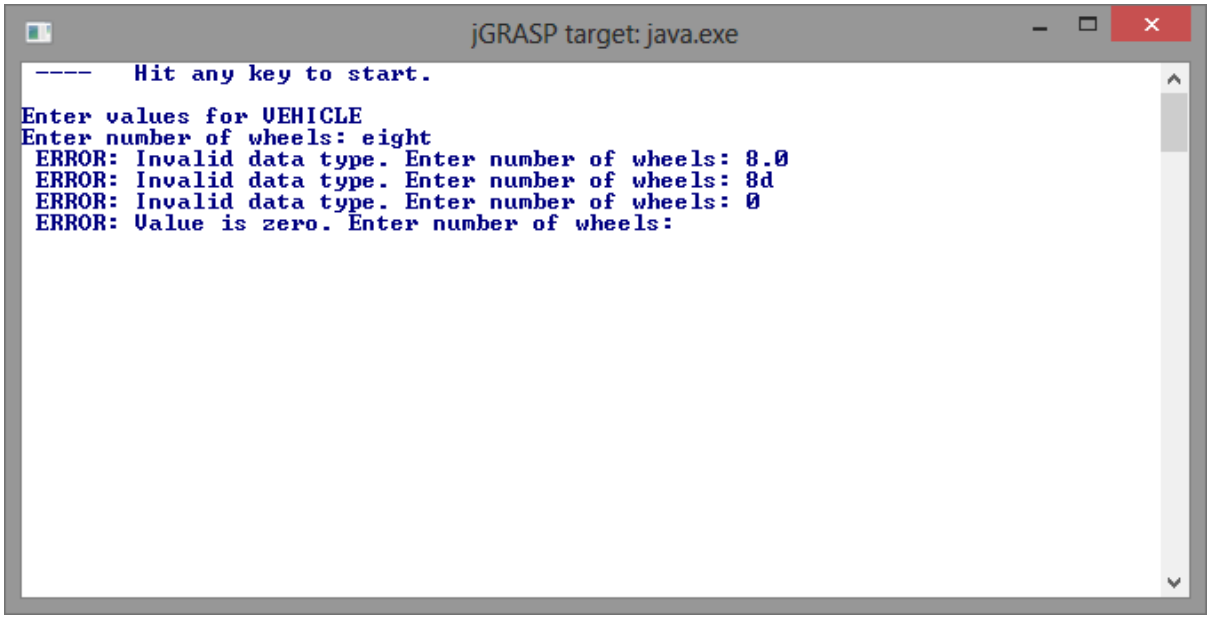
The following section shows images or screenshots of the actual results of the program for functions specified in the table in previous page.

Input values for `Vehicle` object

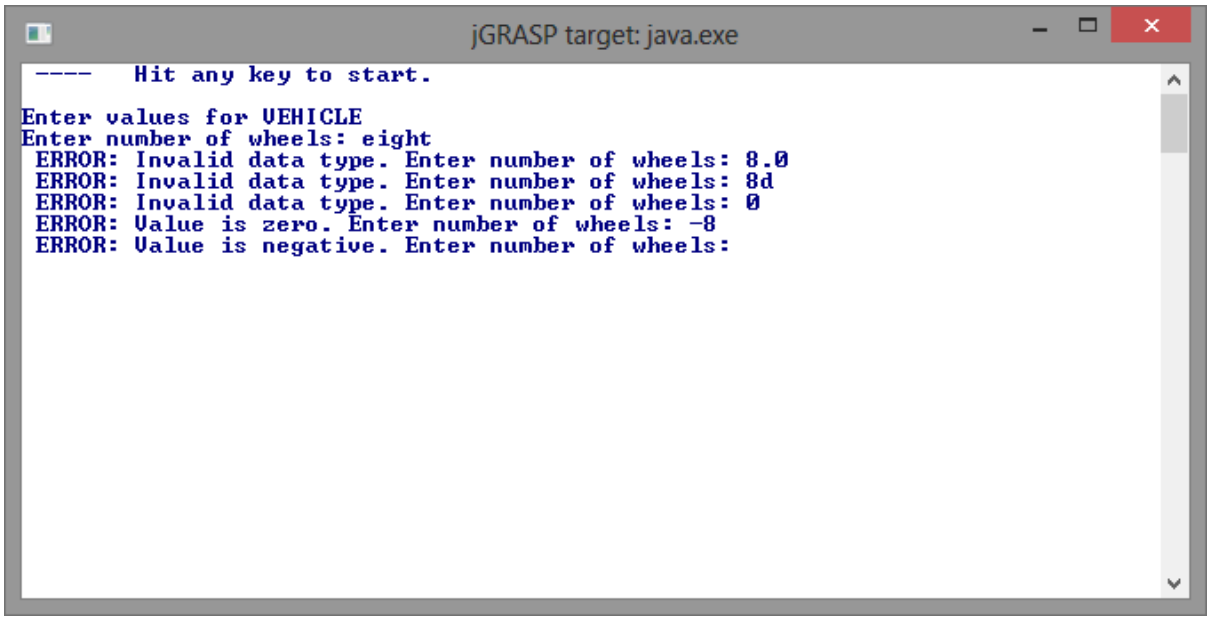
`Vehicle` class requires two values: number of wheels, and average miles per gallon. Program starts by asking user to enter number of wheels for the vehicle object.



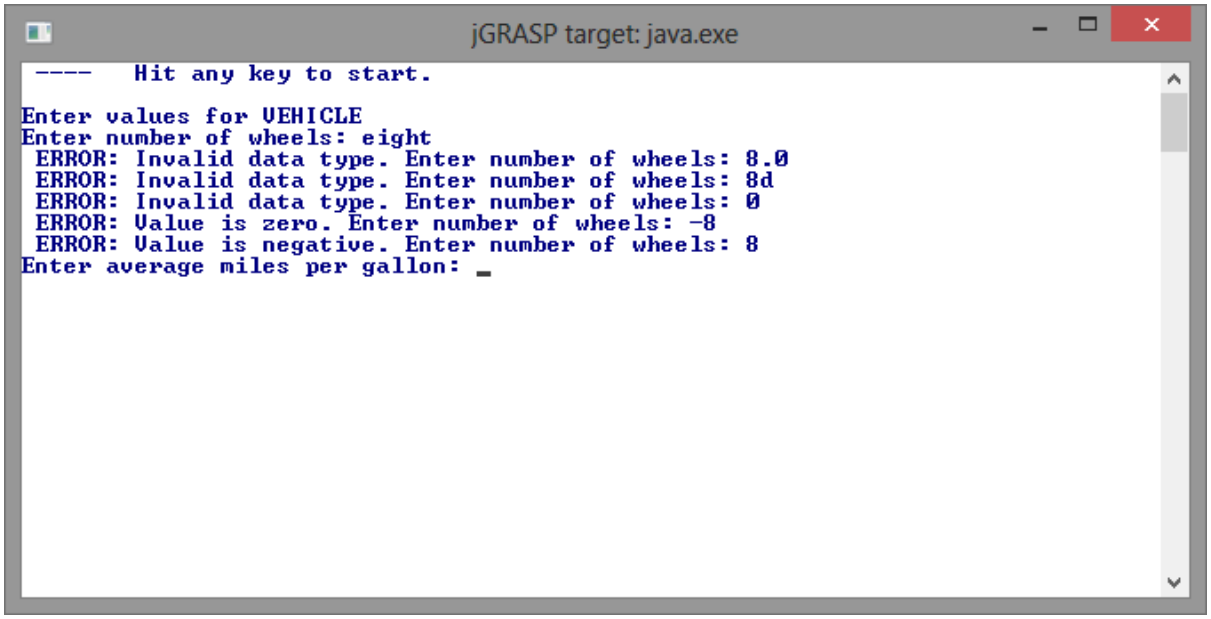
If the user attempted to enter String, double, or any value with symbols and characters, an error message should appear and the program should ask the user again to enter new and valid data.



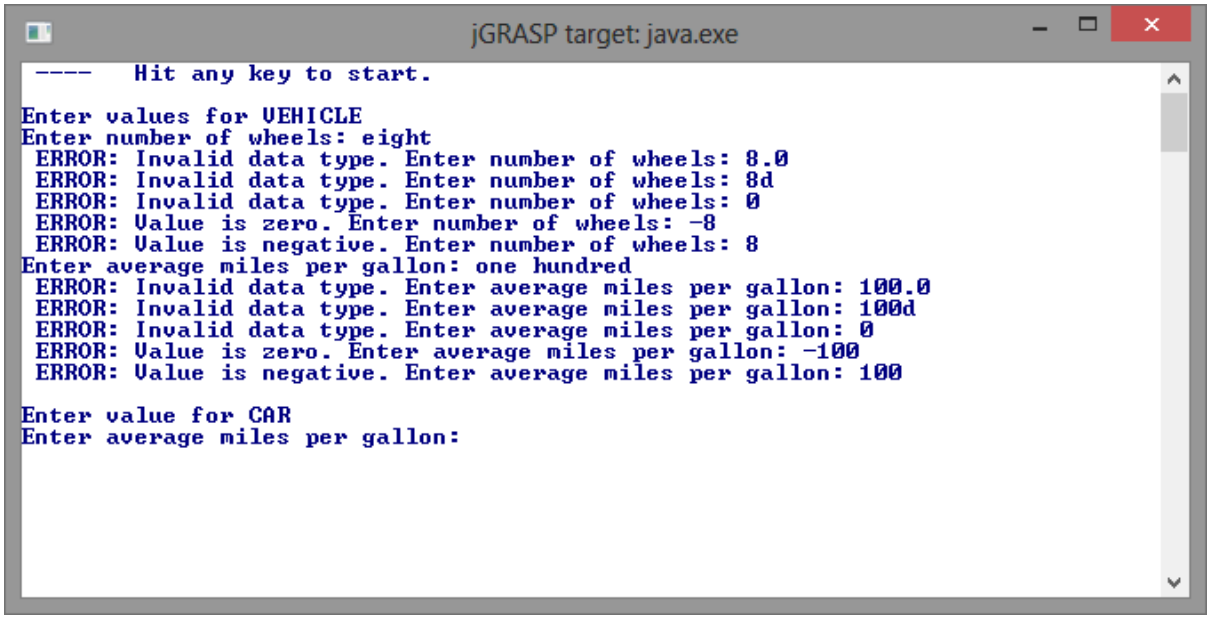
If user attempted to enter zero, program accepts the integer value. However, zero should not be accepted as a value for number of wheels, and is still invalid. Thus, another error message should appear and the program should ask the user again to enter new and valid data.



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.



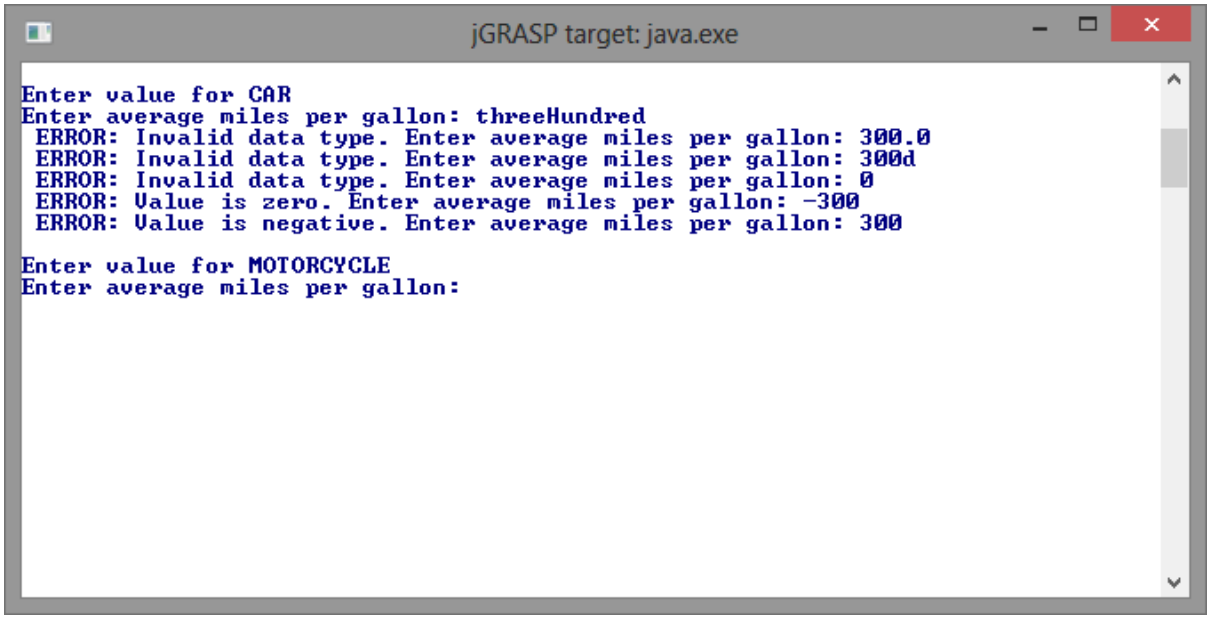
Once the user entered a valid input, program accepts the value and then asks user to input average miles per gallon for vehicle object.



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

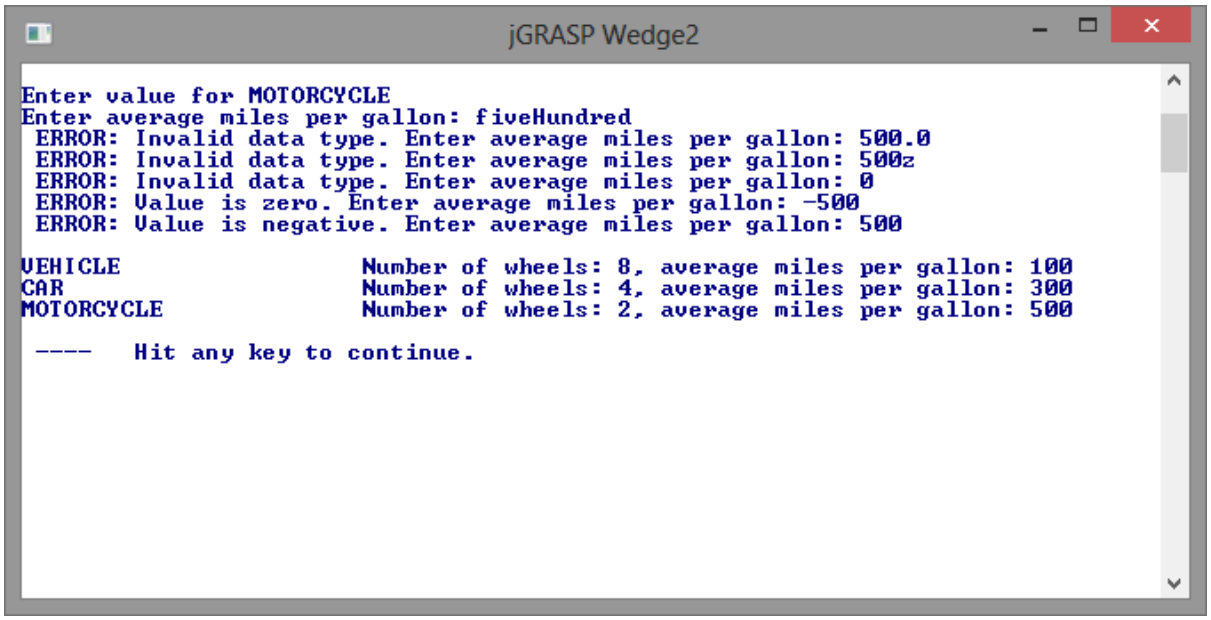
`Car` class requires only one value: average miles per gallon.



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.

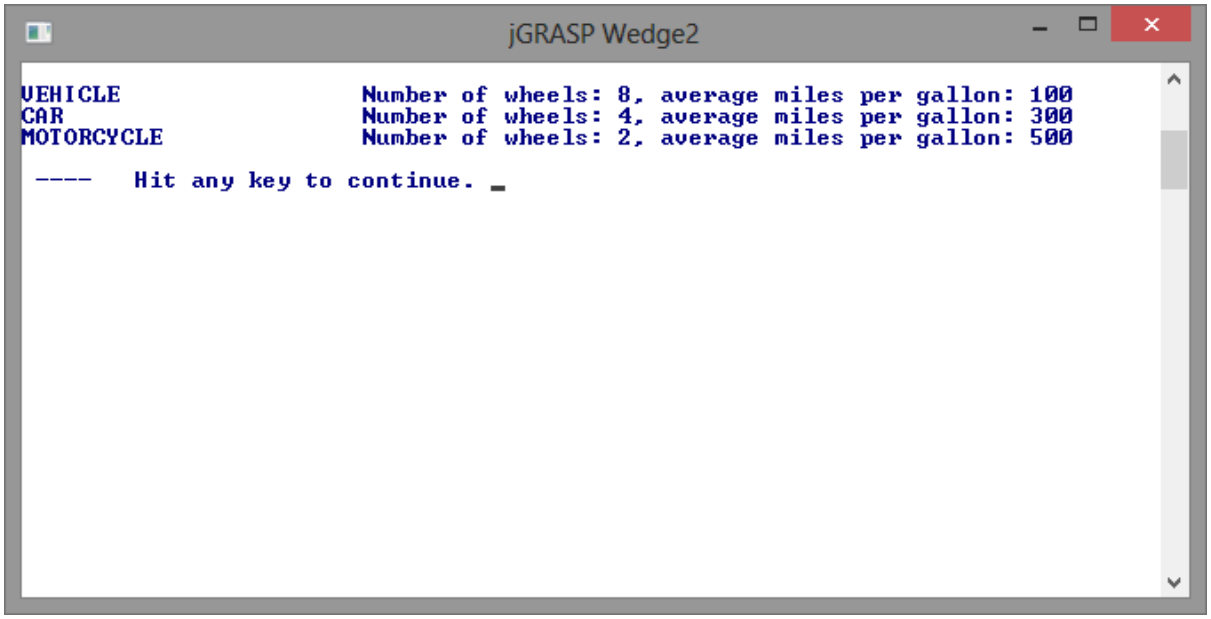
Input values for `MotorCycle` object

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



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



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.