**Assignment 5**

Lab 5  
  
Now we need to modify the program so that we can add additional car models easily and efficiently in a client-server setting.  
  
So let's develop a client-server system where the end users, mostly non-programmers, can easily add new car models to the system. I recommend that you use a text file to store the car data, and then change your previous code to read the data from the text file. The text file is easier for the end users to handle.  
  
Your might consider going through the following steps to do this assignment:

1. Design and create a class called BuildCarModelOptions that can:
   1. Use existing Car or Automotive class, OptionSet and Option Classes to create a car model.
   2. Run inside a Java ServerSocket
2. Design and create a class called CarModelOptionsIO that can
   1. Read data from a Properties Object for building a car model.
   2. Transfers the read data using a Socket Class to the Server.
   3. Receive a response from the Server verifying that the Car Model object is created successfully.
3. Design and create a class called SelectCarOption that can:
   1. Prompt the user of available models
4. Tip: Use Socket Class to interact with Server to find the available models.
5. Allow user to select a model and enter its respective options
6. Display the selected options for a class

Your deliverable for this part:  
   1. Updated classes  
   2. Test program showing the successful implementation of these classes  
  
Tips:  
  
How to read data from a  text file using a Properties Object?  
  
It is convenient to use a properties file to specify the model information. It might contain the following information:  
  
CarModel=Prius  
CarMake=Toyota  
Option1=Transmission  
OptionValue1a=Manual  
OptionValue1b=Automatic  
Option2=Brakes  
OptionValue2a=Regular  
OptionValue2b=ABS  
  
Here is the code for reading a properties file and reads into some local variables.  
  
     Properties props= new Properties();  
     FileInputStream in = new FileInputStream(filename);  
     props.load(in);  
  
     String CarMake = props.getProperty(“CarMake”);  
     if(!CarMake.equals(null))  
      {  
            String CarModel = props.getProperty(“CarModel”);  
            String Option1 = props.getProperty(“Option1”);  
            String OptionValue1a = props.getProperty(“OptionValue1a”);  
            ……  
      }  
  
To learn about creating a Socket in its own thread refer to the Networking Slideshow presented in class.