Create a Car class in Java.

The Car class will have 3 attributes: make (String), model (String) and currentSpeed (an Integer, remember to use int in Java)

The Car class will have 2 constructors: the default constructor will set the make to " " and the model to " ".  Set the speed to 0.  The overloaded constructor will set the make and model to the parameters and the currentSpeed to the matching parameter if the currentSpeed is greater than 0, otherwise set the speed to 0.

The Car class will have accessor and mutator methods for all attributes.  The Car class will also have one method called isSpeeding() that will return a boolean value.  True if the car is speeding, false otherwise.

Create an application class that will instantiate two cars.

Car  car1 = new Car();  //use the default constructor for the first car

Car   car2 = new Car("Ford", "Focus",  100);  //use the overloaded constructor for the second car

Within the main method display all information for the two cars using System.out.println() method calls.

This problem looks very similar to a practice problem.

/\*\*  
 \* Car class  
 \*   
 \*/  
public class Car  
{  
 /\*\*  
 \* declaration for make as a string  
 \*/   
 private String make;  
  
 /\*\*  
 \* declaration for model as a string  
 \*/   
 private String model;  
  
 /\*\*  
 \* declaration for current speed as integer  
 \*/   
 private int currentSpeed;  
  
 /\*\*  
 \* Default constructor   
 \*/

public Car()  
 {  
 make = “ “;  
 model = “ “;  
 average = 0;  
 return;  
 }  
  
 /\*\*  
 \* Overloaded constructor   
 \*/  
 public Student(String newMake, String newModel)  
 {  
 setMake(newMake);  
 setModel(newModel);  
 calcCurrentSpeed();  
 return;  
 }

/\*\*  
 \* the make method returns the value of the make attribute  
 \*/  
 public String getMake()  
 {  
 return make;  
 }  
  
 /\*\*  
 \* the make method gives a value to the make data member  
 \*/  
 public void setMake(int newMake)  
 {  
 make = newMake;

return;  
 }

/\*\*  
 \* the model method returns the value of the model attribute  
 \*/  
 public String getModel ()  
 {  
 return model;  
 }  
  
 /\*\*  
 \* the model method gives a value to the model data member  
 \*/  
 public void setModel (int newModel)  
 {  
 model = newModel;

return;  
 }  
  
 /\*\*  
 \* the getCurrentSpeed method returns the currentSpeed   
 \*/  
 public int getCurrentSpeed( )   
 {  
 return currentSpeed;  
 }  
   
 /\*\*  
 \* The calcCurrentSpeed method returns a matching parameter if currentSpeed is greater than 0 else it sets it to 0  
 \*/  
 public void calcCurrentSpeed( )   
 {  
 if (currentSpeed > 0)

currentSpeed = newCurrentSpeed;

else

currentSpeed = 0  
 return;  
 }

/\*\*  
 \* The isSpeeding method is a boolean that evaluates to true or false

\*/  
 public void isSpeeding( )   
 {  
 if currentSpeed > 80

Set isSpeeding = true

else

isSpeeding = false  
 return;  
 }  
  
   
 /\*\*  
 \* the displayGrade method displays an appropriate letter grade   
 \*/  
 public void displayGrade( )   
 {  
 System.out.println ("Test 1 score is: " + test1);   
 System.out.println ("Test 2 score is: " + test2);   
 if (average >= 90)  
 System.out.println("A");  
 else if (average >= 80)  
 System.out.println("B");  
 else if (average >= 70)   
 System.out.println("C");  
 else if (average >= 60)   
 System.out.println("D");  
 else   
 System.out.println("F");  
 }  
   
}

/\*\*  
 \* instantiates two cars following the instructions:

\*/  
 public static void main(String[] args)   
 {  
 Car car1 = new Car();   
 Car car2 = new Car(“Ford”, “Focus”, 100);  
   
 car1.displayMake();  
 car1.displayModel();  
 car1.displayCurrentSpeed();  
 car1.displayIsSpeeding();

car2.displayMake();  
 car2.displayModel();  
 car2.displayCurrentSpeed();  
 car2.displayIsSpeeding();  
  
 System.out.println();  
   
 return;   
 }  
}