AP Java Car class Lab
To Create and Use Classes

==========

Objectives:
- To design and write a classes
- To create objects from classes.

Show me:
- `Car.java` file
- `TestVehicle.java` file
- output

==========

Directions:

1. Create a class called `Car`. The `Car` class should have the following items:

a. 2 private instance variables

Variable name	*Description*	*Datatype*
`color`	Color of the car such as "black", "white", etc.	`String`
`speed`	The car's current speed	`double`

b. 2 constructors

Constructor	*Parameters*	*Purpose*
No-argument constructor	none	Set the initial value of `color` to the default value of "`black`" and the initial value of `speed` to the default value of `0.0`.
2 argument constructor	`String inColor` `double inSpeed`	Initialize the `color` and `speed` to the values passed in.

c. 8 public methods

Method name	*Parameters*	*Return Type*	*Purpose*
`getSpeed`	none	`double`	Accessor method for `speed` - returns the current value of the private variable `speed`.
`getColor`	none	`String`	Accessor method for `color` – returns the current value of the private variable `color`
`setSpeed`	`double inSpeed`	none	Mutator method for `speed` – sets a new value for the car's speed
`setColor`	`String inColor`	none	Mutator method for `color` – sets a new value for the car's color
`go`	none	none	Sets the car's speed to `60` and displays a message that says "The `car is moving with speed of s`" (replace s with the current speed).
`go`	`double inSpeed`	none	Sets the car's speed to a value passed in and displays a message that says "`The car is moving with speed s`" (replace s with the current speed).
`stop`	none	none	Sets the car's speed to `0` and displays a message that says "`The car is not moving. The speed is s`" (replace s with the current speed).
`toString`	none	`String`	Returns "`I am a car with speed s and color c`" (replace the s with the car's current speed and the c with the car's current color).

2. Create a class called `TestVehicles` that does the following in the `main` method, in the order specified:

a. Declare a variable called `slowCar` of type `car` and instantiate a `Car` object using the no-argument constructor.

b. Declare a variable called `fastCar` of type `car` and instantiate a `Car` object using the correct constructor to set the speed to `100` and the color to `red.`

c. Using the `slowCar` do the following:

i. Use the accessor method to print out the car's current speed.

ii. Use the accessor method to print out the car's current color.

iii. Use the mutator method to change the speed to `45`.

iv. Use the mutator method to change the color to `green`.

v. Use the accessor method to print out the new value of the car's speed.

vi. Use the accessor method to print out the new value of the car's color.

vii. Add the following line of code: (This line calls the `toString()` method)
```
System.out.println(slowCar);
```

viii. Make the car go with the default speed by calling it's `go()` method.

ix. Make the car go with a new speed of `50` by calling it's `go(double inSpeed)` method.

x. Make the car stop by calling it's `stop()` method.

d. Using the `fastCar` do the following:

i. Use the accessor method to print out the car's current speed.

ii. Use the accessor method to print out the car's current color.

iii. Use the mutator method to change the speed to `0`.

iv. Use the mutator method to change the color to `yellow`.

v. Use the accessor method to print out the new value of the car's speed.

vi. Use the accessor method to print out the new value of the car's color.

vii. Add the following line of code: (This line calls the `toString()` method)
```
System.out.println(fastCar);
```

viii. Make the car go with the default speed by calling it's `go()` method.

ix. Make the car go with a new speed of `120` by calling it's `go(double inSpeed)` method.

x. Make the car stop by calling it's `stop()` method.