**TEST CASE STRATEGY FOR CAR CLASS**

**CLASS DETAILS FOR CAR CLASS**

Attributes

private String regno;

private int yearmade;

private String[] colour;

private String carmake;

private String carmodel;

private int price;

**Constructor**

Car()

**Accessor methods**

getColour(): String[]

getCarmake(): String

getCarmodel(): String

getPrice(): int

getRegNo(): String

getYearmade(): int

**Mutator methods**

setColour(String[])

setCarmake(String)

setCarmodel(String)

setPrice(int)

setRegNo(String)

setYearmade(int)

TEST PLAN

### Create Car with default constructor

### Test get methods (accessors)

### Test get methods (accessors) and mutators with

### With valid input field values

##### With invalid input field values

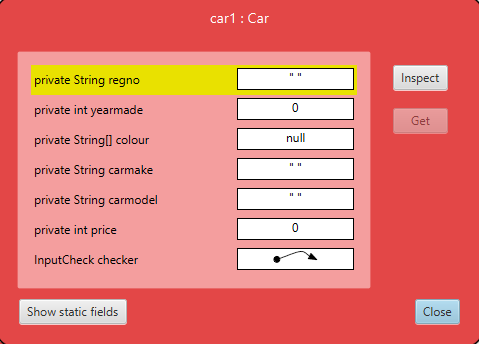
**TEST 1- Testing the default constructor**

Test input data - NO INPUT

**Expected output-**

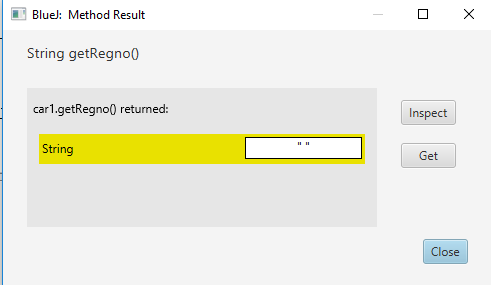
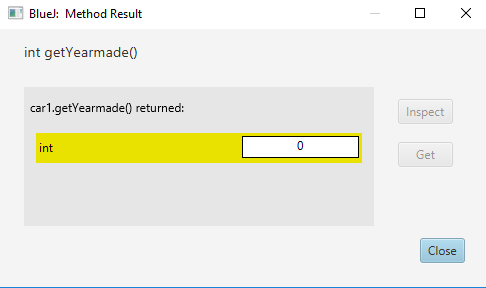
* Regno = “”
* Yearmade = 0
* Colour = null
* Carmodel = “ “
* Carmake = “ “
* Price = 0

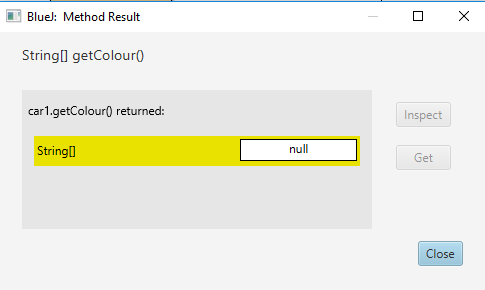
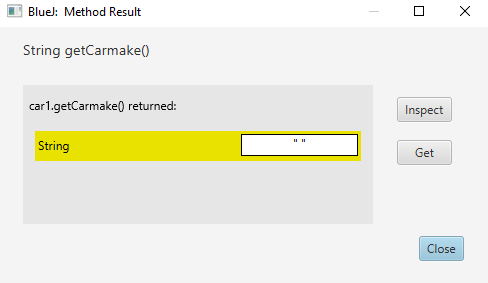
**Actual Result**

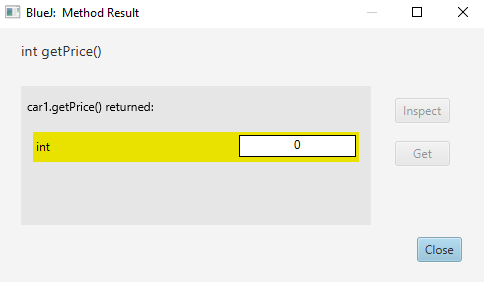
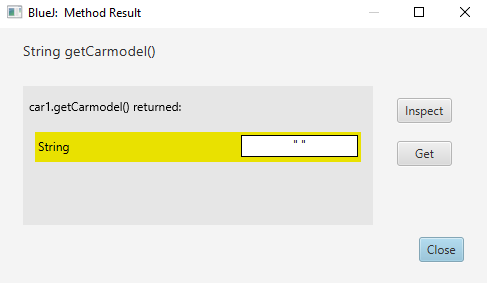
****

**TEST VERIFIED**

**2.TEST 2 – VERIFYING THE GET METHODS TO CHECK THE INITIAL VALUES THAT HAVE BEEN SET**

TEST VERIFIED – ALL GET METHODS SHOW THE CORRECT INITIALISED VALUES

**3-TESTING EACH SET AND GET METHOD USING DATA**

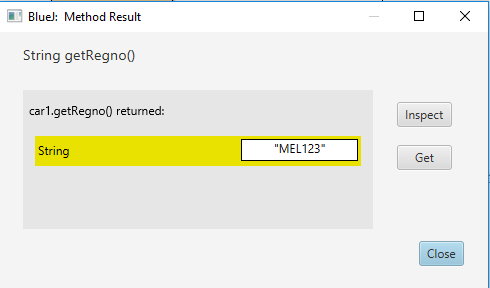
3 -Testing car registration number using valid input

Test data

“MEL123”

EXPECTED RESULTS

“MEL123”



TEST VERIFIED

3 -Testing car registration number using invalid input of length greater than 6

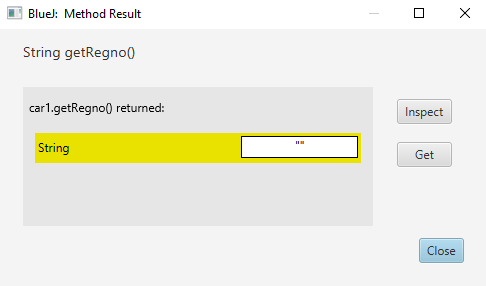
Test data

“MEL12356”

EXPECTED RESULTS

“”

ACTUAL RESULT



The input does not meet the length being 6 or less

TEST VERIFIED

3 -Testing car registration number using invalid input containing characterrs like ..?\*

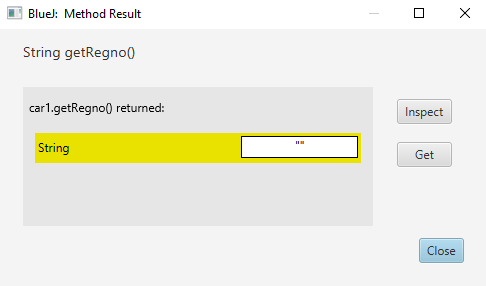
Test data

“AD??\*”

EXPECTED RESULTS

“”

ACTUAL RESULT



TEST VERIFIED

3 -Testing car registration number using invalid input containing spaces

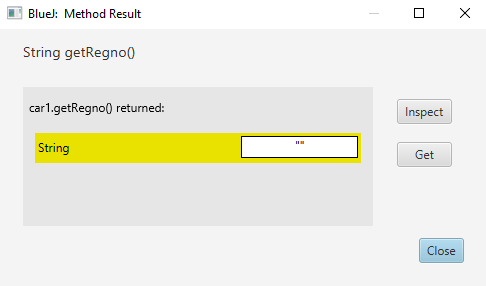
Test data

“MEL 78”

EXPECTED RESULTS

“”

ACTUAL RESULT

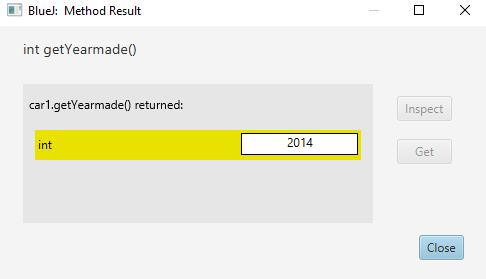


TEST VERIFIED

3-Testing the year with valid input

Test data-2014

Expected output- 2014

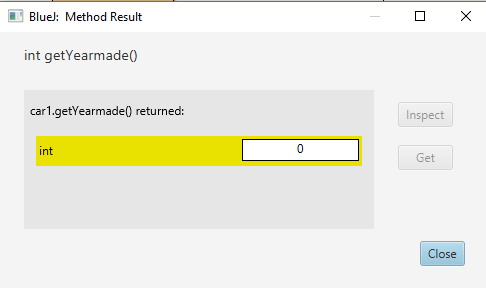


TEST VERIFIED

3-Testing the year with invalid input like 3456 which is not a year

Test data-3456

Expected output- 0

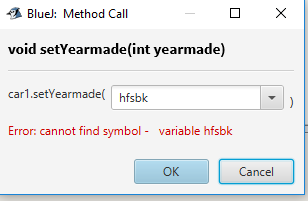


TEST VERIFIED

3-Testing the year with invalid input like ghsf which is not numeric

Test data-bghy

Expected output- 0

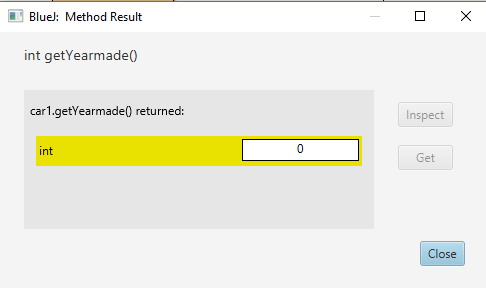


TEST VERIFIED

3-Testing the year with invalid input like 2018 which is which greater than 2017

Test data-bghy

Expected output- 0

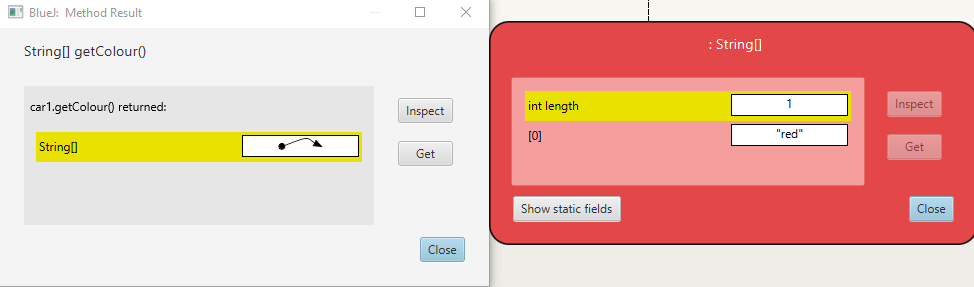


TEST VERIFIED

3-Testing the colour with a valid input Test data

Test data-{“red” }

Expected output--{“red”}

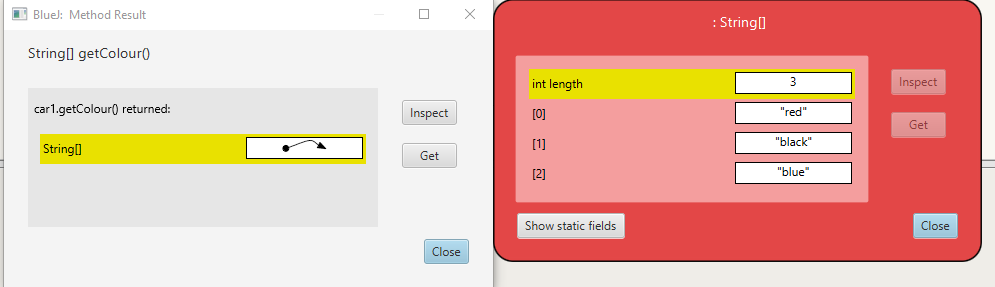


TEST VERIFIED

3-Testing the colour with a valid input Test data of length 3

Test data-{“red”,”blue”,”black”}

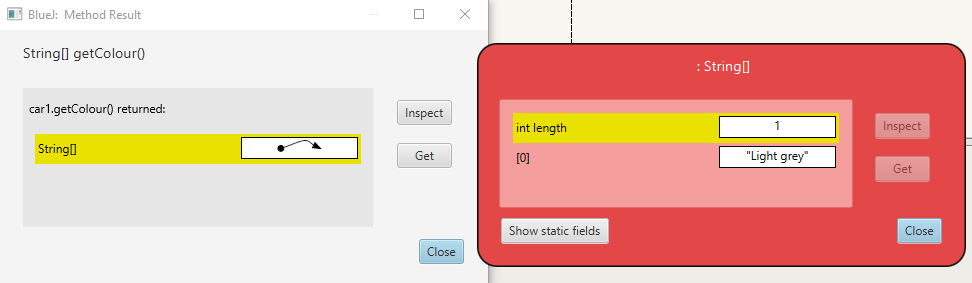
Expected output--{“red”,”blue”,”black”}



3-Testing the colour with a valid input Test data with space

Test data-{“light grey”}

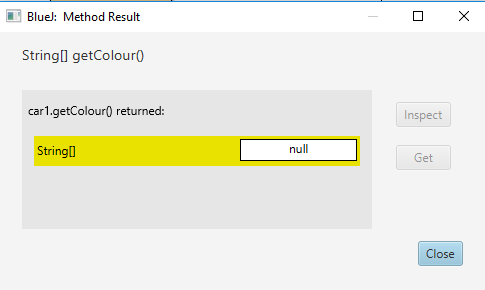
Expected output--{“light grey”}



3-Testing the colour with a invalid input Test data with numeric input

Test data-{“Light grey123”}

Expected output--{“”}

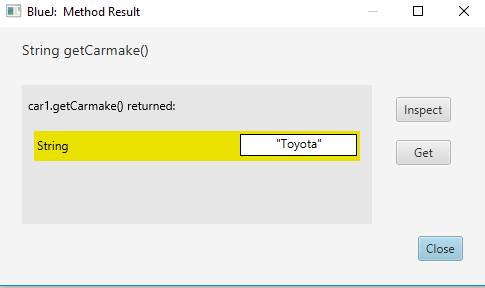


TEST VERIFIED

3-Testing the carmake with a valid input

Test data-{“Toyota”}

Expected output--{Toyota}

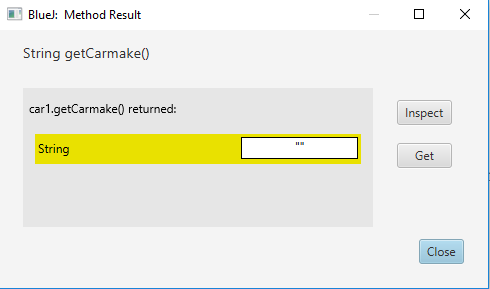


TEST VERIFIED

3-Testing the carmake with a valid input

Test data-{“Toy ota”}

Expected output--{“” }

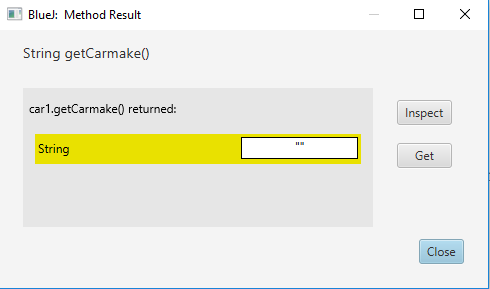


TEST VERIFIED

3-Testing the carmake with a valid input

Test data-{“??LL”}

Expected output--{“” }

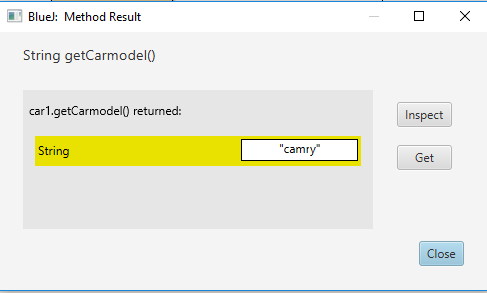


TEST VERIFIED

3-Testing the carmodel with a valid input

Test data={“Camry”}

Expected output--{Camry}

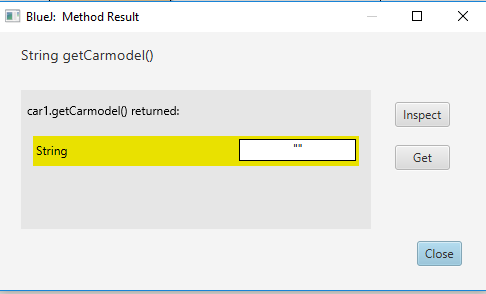


TEST VERIFIED

3-Testing the carmodel with a invalid input

Test data={“Cam ry”}

Expected output--{“”}

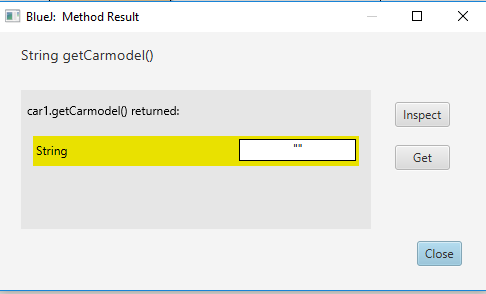


TEST VERIFIED

3-Testing the carmodel with a invalid input

Test data={“Cam??ry”}

Expected output--{“”}

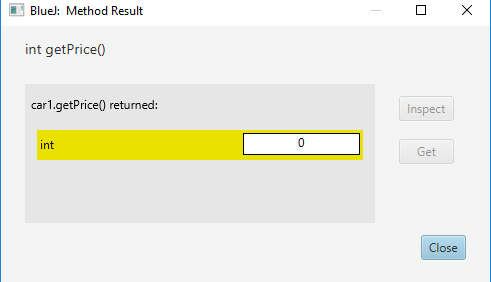


TEST VERIFIED

3-Testing the price

Test data={“-56}

Expected output--{0}

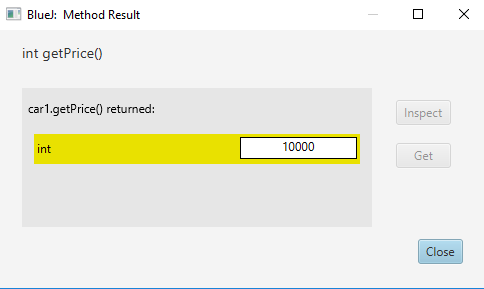


TEST VERIFIED

3-Testing the price

Test data={10000}

Expected output--{10000}



TEST VERIFIED