

Lab 1: Basic JUnit 4

Software Engineering Prof. Bradford

Due date: See HuskyCT

Objective

To use Junit 4 to do basic unit tests.

Preparation

Using Git Codespaces

Create folder structures: Lab-1

```
CodeSpace> cd Lab-1
CodeSpace> mkdir src
CodeSpace> cd src
CodeSpace> mkdir c u
```

```
// c is for code
// u is for unit tests
```

Details

Put your Java class TestMe.java into **src/c**

Put your Java class TestMeTest.java into **src/u**

In the folder above src (j2024_09_03)

```
CodeSpace> wget https://repo1.maven.org/maven2/junit/junit/4.13.2/junit-4.13.2.jar
CodeSpace> wget https://repo1.maven.org/maven2/org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.jar
```

```
CodeSpace> cd /workspaces/Lab-1/  
CodeSpace> ls  
src/  
junit-4.13.2.jar  
hamcrest-core-1.3.jar
```

ClassPath is the single jar file and the sources are explicitly written out

```
CodeSpace> javac -cp junit-4.13.2.jar "./src/c/TestMe.java" "./src/u/TestMeTests.java"
```

ClassPath is ./src/ so c.TextMe is referenced from ./src/

```
CodeSpace> java -cp "./src/" c.TestMe
```

No public static void main(String[] args) !

How can we fix this?

Need the main from org.junit.runner.JUnitCore to pull the method annotated by @Before then running all functions annotated with @Test
The annotations and test runners are in the jar files

```
CodeSpace> pwd  
/workspaces/Lab1/j2024_09_03
```

```
CodeSpace> java -cp "junit-4.13.2.jar:hamcrest-core-1.3.jar:./src/" org.junit.runner.JUnitCore  
u.TestMeTests
```

Try to remove "./src/" from this last line. What happens?

Note the parts of the ClassPath are separated by ":"

The challenge

Gasoline vehicles efficiency is measured in MPG (miles per gallon) or miles/gallon. So it is easy to measure the efficiency of two different vehicles

Electric vehicles efficiency is measured in MPGe (miles per gallon equivalent). Where 1 gallon of gasoline is equal to 33.7 kWh.

So, if an electric vehicle goes 100 miles per 33.7 kWh, then this vehicle is going 100 MPGe (mpg equivalent). Or 33.7 kWh/1 gallon is an MPGe unit.

Suppose, in Connecticut, residential electricity is 24 cents per kWh.

So, 33.7 kWh times 0.24 dollars/kWh gives a cost of \$8.09/MPGe.

So if an electric car has a trip that goes 120 miles using MPGe = 33.7 kWh/gallon, then the cost of this trip is \$8.09 for 120 miles.

If a gasoline car has MPG = 20 miles/gallon and suppose gasoline costs \$3.50/gallon, then a 120 mile trip requires 120 miles/ (20 miles/gallon) = 6 gallons. So the cost for this 120 mile trip is 6 gallons * \$3.50/gallon = \$21.

What to design and build

Design and build a Gasoline interface and an Electric interface.

```
public interface GasolineInterface {  
    |  
    double calcGasMPG();  
    void setMilesfromGas(double miles);  
    void setGallonsfromGas(double gallons);  
  
}
```

And

```
public interface ElectricInterface {  
  
    double calcMPGe();  
    void setElectricMiles(double totalElectricMiles);  
    void setTotalkWh(double totalkWh);  
  
}
```

A HybridVehicle class should **implement** both interfaces. You should add set/get method stubs for the cost per gallon and cost per kWh in the appropriate interfaces.

A main class CarRunner should instantiate a HybridVehicle class instance. The CarRunner should calculate the following examples:

1. The MPG when in fully gas mode
2. The MPGe when in fully electric mode
3. The average MPG $(\text{MPG} + \text{MPGe})/2$ when in half-gas/half-electric hybrid mode

Assuming gasoline costs \$3.50/gallon and 1 kWh costs \$0.24.

As an example, suppose in fully electric mode, your car travels 300 miles using 70 kWh.

The MPGe is computed as $300 \text{ miles} / 70 \text{ kWh} \times 33.7 \text{ kWh} / 1 \text{ gallon} = 144.43 \text{ MPGe}$.

You must build unit tests for this codebase.

What to hand in

Brief description of what you did (< 1 page)

A video of your builds and **tests** running

All Junit tests for the class Hybrid.

Repo link and the README file must show how to download/build/run your system