BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
K. K. BIRLA GOA CAMPUS
FIRST SEMESTER 2023-2024

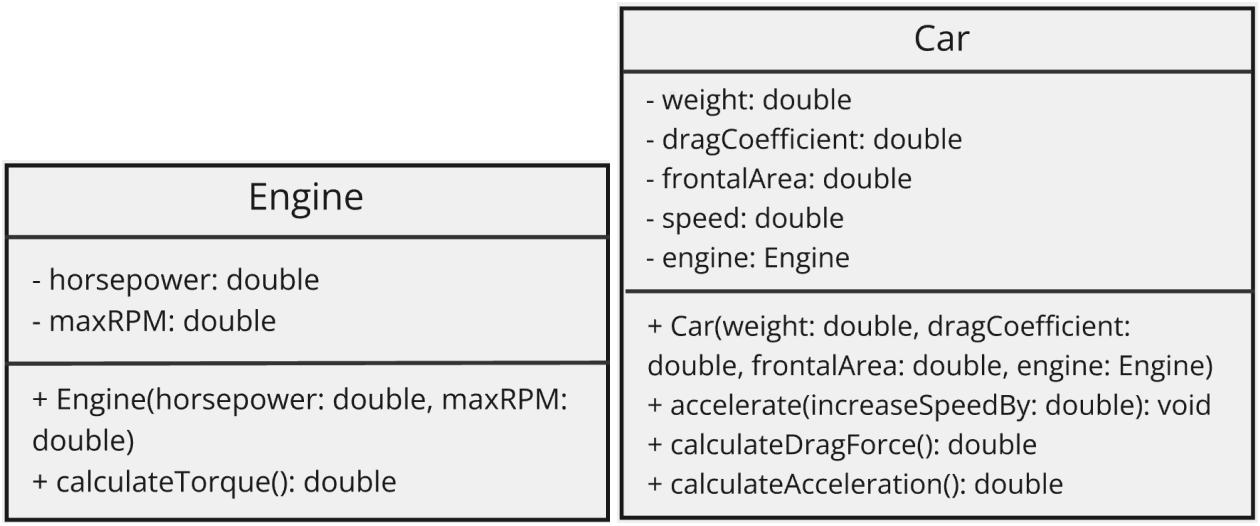
CS F213 (Object Oriented Programming) - Lab 6, Date: 11/09/24

Problem Statement

In this lab assignment, you will implement a car by creating 2 interacting classes - `Engine` and `Car`. The details of each class and its functions are provided in the form of UML diagrams, which you must understand and translate into Java code.

UML Diagrams

The UML diagrams provided below outline the structure and relationships of the classes you need to implement. **You also need to implement getter functions for all private variables in both classes**. The naming of the getter functions should follow camel case. For example – getHorsepower(), getMaxRPM() and getDragCoefficient().



Useful implementations:

1. `double calculateTorque()`: Calculates and returns the engine's torque using the formula:

$$\text{Torque} = \frac{\text{Horsepower} \times 5252}{\text{Max RPM}}$$

2. `double speed`: Stores the current speed of the car (initially set to 0).

3. `void accelerate()`: Increases the speed of the car by the value `increaseSpeedBy.`

4. `double calculateDragForce()`: Calculates and returns the drag force acting on the car using the formula:

$$\text{Drag Force} = 0.5 \times \text{dragCoefficient} \times \text{frontalArea} \times \text{speed}^2$$

5. `double calculateAcceleration()`: Calculates and returns the car's maximum acceleration based on the engine's torque using the formula:

$$\text{Acceleration} = \frac{\text{Torque} \times 746}{\text{Weight} \times 9.81}$$

Example:

```
Engine engine1 = new Engine(400, 6000);
Car car1 = new Car(1500, 0.3, 2.2, engine1);

Expected Output:
Car 1 Torque: 350.1333333333333
Car 1 Drag Force: 825.0
Car 1 Acceleration: 17.75055838713331
```

TestCase1:

Program Input:
1,100,1000
Program Output:
100.0
1000.0

TestCase2:

Program Input:
1,100,1000
Program Output:
100.0
1000.0

Instructions:

Follow the steps given below to complete the OOP lab problem.

Step 1: Read the Lab Question

Read and understand the Lab problem given above.

Step 2: Give execute permission to the executable files

Use the following command to give execute permission to the executables.

```
:~$ chmod +x RunTestCase CreateSubmission
```

Step 3: See the input and the expected output

Use the following command to see the input and the expected output for test case `T1`. Note that `L1` refers to Lab 1 (use the correct lab number), `Q1` refers to Question 1 and `T1` refers to test case 1.

```
:~$ ./RunTestCase L1 Q1 YourBITSId T1
```

Use your own (13 character) BITS Id in place of `YourBITSId` in the above command. Type your BITS Id in upper case (capital letters). Ensure that you enter your BITS Id in 202XA7PSXXXXG format.

Run the command from within the folder containing the executables and the java files.

Step 4: Modify the solution java file

Modify the solution Java file(s) to solve the question given above. Repeat Step 3 until all the test cases are passed. The test cases are numbered `T1`, `T2`, etc.

Step 5: Passing evaluative test cases

There are evaluative test cases whose expected output is hidden. The evaluative test cases are numbered `ET1`, `ET2`, etc. The lab marks will be based on the evaluative test cases. Use the following command to check whether your solution passes the evaluative test cases.

```
:~$ ./RunTestCase L1 Q1 YourBITSId ET1
```

Ensure that your *final* solution passes all the evaluative test cases `ET1`, `ET2`, etc. The expected output of the evaluative test cases are *hidden*.

Step 6: Create submission zip file

Use the following command to create the submission zip file.

```
:~$ ./CreateSubmission L1 YourBITSId
```

The first command line argument must correspond to the lab number and the second command line argument must be your 13 character BITS id.

The above command will create a zip file. The command will also list the evaluative test cases that your solution program has passed.

Step 7: Upload submission zip file

Upload the submission zip file created in Step 6. Do not change the name of the zip file or modify any file inside the zip file. You will not be awarded marks if the zip file is tampered with in any manner.