Question 1: (try-catch-finally) 10 Marks

Problem description: Create a calculator application providing following methods:

1. Add
2. Subtract
3. Multiply
4. Divide

Apply appropriate exception handling to the methods wherever required

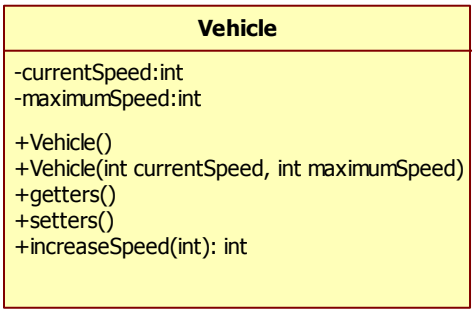
Test Case ID	Input	Output	Marks
UTC01_01	Add(5,10)	15	10
UTC01_02	Subtract(50,25)	25	
UTC01_03	Multiply(28,7)	196	
UTC01_04	Divide(45,5)	9	
UTC01_05	Divide(45,0)	Divide by Zero exception	

Question2 (Custom exception) 10 Marks

Problem description:

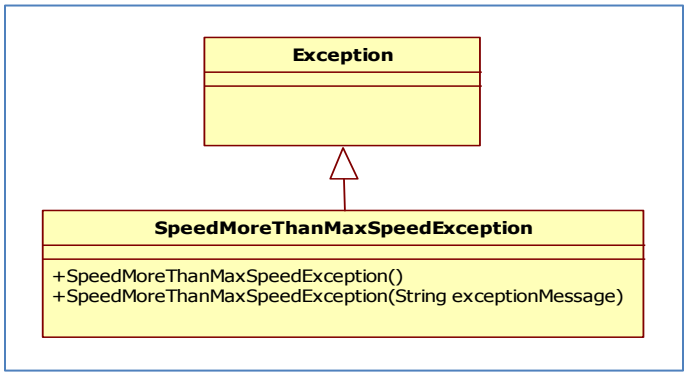
A vehicle has current as well as maximum speed and when the current speed of that vehicle is increased, the current speed might reach the maximum speed at one time. Once the vehicle's current speed reaches maximum speed, a user defined exception should be thrown by the application and should be handled by user.

Write a class Vehicle as shown in the class diagram below:



Call IncreaseSpeed() method to increase the speed of the Vehicle and returns updated current speed. If the current speed crosses maximum speed, a user defined exception should be thrown.

Write a custom exception class as given in the class diagram:



Test Case ID	Input	Output	Marks
UTC02_01	MaximumSpeed=20 ,CurrentSpeed=10, IncreaseSpeed=5	15	5
UTC02_02	MaximumSpeed=30 ,CurrentSpeed=10, IncreaseSpeed=20	30	
UTC02_03	MaximumSpeed=30 ,CurrentSpeed=10, IncreaseSpeed=25	SpeedMoreThanMaxSpeedException	5

Question3 (Check Numbers) 10 Marks

Problem description: Given a string array of valid and invalid integer and decimal numbers. Complete the provided method which accepts a String array of valid and invalid numbers and check if the numbers are Valid or invalid Integer or decimal number.

Input: A String array of Valid and invalid integer and decimal numbers

Output: The output is a list of strings that gives the message <Number> is a valid decimal number or <Number> is an invalid decimal number or <Number> is a valid Integer number or <Number> is an invalid integer number

Implement above scenario and create Junit test methods for the below cases:

Test Case	Input	Output	Marks
UTC3_ 01	10.29, 12345, 12.0.0	10.29-valid-decimal, 12345-valid-integer, 12.0.0-invalid-decimal	5
UTC3_02	Number, 12	Number-invalid-integer, 12-valid-integer	
UTC3_03	Number.0, 12a	Number.0-invalid-decimal, 12a-invalid-integer	5
UTC3_ 04	2.0a, 10,0.5	2.0a-invalid-decimal, 10-valid-integer, 0.5-valid-decimal	