|  |  |  |
| --- | --- | --- |
|  | ***EUROPEAN UNIVERSITY OF LEFKE***  ***FACULTY OF ENGINEERING***  *Computer Engineering Department*  ***COMP471 Java Programming***  *Asst.Prof.Dr.Vesile Evrim* |  |
|  | ***2019/2020 Summer Semester Final***  ***Due (20/8/2020 15:00 )*** |  |

**The exam/project has been submitted by myself is solely my own work and I did not get any aid from other people.**

**Name: ……………………….**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
|  | Maximum | Received |
| Problem 1 | 100 |  |
| Total | 100 |  |

**NOT:** Please read the **RULES** at the end of the question before solving the question and also the **SUBMISSION GUIDELINES**.

**Problem 1:** You are requested to implement the code for the given UML diagram. It means you might not have said directly that you need to define a variable or implement a function but if it is given in the UML diagram and need to be implemented please do so. Some of the additional details are provided below. **(read the rules at the end of question before start solving it)**

Vehicle Transport <Interface> Constraction<interface>

Weight:int

Height:double

Fee:int

GetWeight():int

SetW-fineight(weight:int):void

GetHeight():double

SetHeught(height:double):void

RegistrationFee():double

GetMaterial():ArrayList<String>

Compose():String

City:String

Fare():double

Car Truck

Color:String

Make:String

Motor: int

CMaterial:String



SetMotorPower(motor:int):void

GetMotorPower (): int

Capacity:int

Year: int

Material: ArrayList<String>



SetCapacity(j:int):void

GetCapacity (): int

**NOT**: For each .java file please put “// stno\_name\_surname” line on top of your file.

1. Create an abstract “Vehicle” class, which is the abstract superclass of “Car” and “Truck” classes. (15 pts)
2. Declare a constant “Fee” variable with value 5, which is fix and cannot be changed.
3. Declare and abstract method “RegistrationFee”
4. The constructor of Vehicle assigns a value to the “Height” and “Weight” variables
5. Create the “Transport” interface as specified by the UML diagram. (5pts)
6. Create the “Construction” interface as specified by the UML diagram. (5 pts)
7. Create “Car” class which is a subclass of a Vehicle class (20pts)
   1. Note that the constructor of a Car class set the values for “Color”, “Make” and “Motor” variables.
8. Create “Truck” class which is a subclass of a Vehicle class. Notice that Truck Implements Transport and Construct interfaces. (25 pts)
   1. In the constructor of a Truck class, you are required to add the material (e.g., sand, gravel) which can be carried by the truck to the Materials list of the class.
9. Create TestVehicle.java class which will have the following information in it. Please do not erase the comments in your code and obey the RULES mentioned below (30 pts)

// Create a red Honda with a motor power, 20 use variable name c1 for the object

//create c2 object of a car which is 1600 kg and 2-meter-high in type of Vehicle

// Create a Truck t1 which can carry “sand”

//Create t2 object of a truck which carries “gravel” so that it allows you to only print the city and calculate fare but not able to set the capacity

//print the registration fee for c2 object.

//print the material carried by t1 object

// assign the weight of t1 object as 3000

//print the weight of the t1 object

//print the city information of t2 object

**Information about functions:**

1. In general, all the functions which start with “set” assign the value to the related variable and the ones stars with “get” return value of the related variable.
2. “RegistrationFee”, this method is used to calculate the fee you need to pay which is the multiplication of the Weight , Fee and type of the vehicle( type value for car is 1, for truck it is 2).
3. “Fare” function calculate the amount should be paid which is based on the weight and the age of the vehicle.

**Rules about classes:**

-You cannot define any new function other than available ones in UML diagram

-If it is needed you can **only add one extra** constructor to the classes which should not take more than 2 parameters.

-You should use just necessary functions in each class. Adding extra functions other than the ones in UML or the ones enforced by object oriented rules, won’t be accepted.

-If you are asked to have constants, you can assign a value to constants.

- you can’t write any “print statement” out of the TestVehicle.java class

-you cannot directly print the value in any print statement ex:“system.out.println(“sand”) is not accepted.

-you cannot create any other object than the stated in the TestVehicle.java class.

**Submission Instructions:**

**1-Make sure to fill the first page of this file.**

Submission without the name, id, and acceptance of statement ( **you can put your name**) in the first page will not be accepted

You are requested to place your answers (code in each file) at the end of this file. Have a separate page and paste your code as (ex:

//Vesile Evrim

Car.java

Public class…..

3- Please read the question carefully. You are responsible from the deducted points if you do not obey the rules as explained in the above question

4- Your code is going to be checked for plagiarism, so please do it yourself.

5- In addition to final.doc document you are requested to submit your code as a project. Go to workspace and .rar or .zip your project from there

6-You need to submit this exam electronically through the reserved place in moodle by **20/8/2020 15:00 sharp.** **No extension will be provided for the exam and the system will be closed for submissions after 15:00. Submissions from any other environment and/or another time will not be accepted.**

You should submit studentno\_comp471final.doc document together with your “project” folder from workspace as

**studentno\_name\_surname\_comp471\_final.zip**

**Good Luck,**

**Vesile**