

**1st Sit Coursework 1 Question Paper:****Year Long 2021/2022**

<b>Module Code:</b>	<b>CS4001NP</b>
<b>Module Title:</b>	<b>Programming</b>
<b>Module Leader:</b>	<b>Sushil Paudel</b>

<b>Coursework Type:</b>	<b>Individual</b>
<b>Coursework Weight:</b>	This coursework accounts for <b>30%</b> of your total module grades.
<b>Submission Date:</b>	<b>12<sup>th</sup> Week</b>
<b>When Coursework is given out:</b>	<b>8<sup>th</sup> Week</b>
<b>Submission Instructions:</b>	<p>Submit the following to Informatics College Pokhara RTE department before the due date:</p> <ul style="list-style-type: none"><li>• <b>A report in PDF format and zip file which includes program file</b></li><li>• <b>File should be in .java format</b></li></ul>
<b>Warning:</b>	London Metropolitan University and Informatics College Pokhara takes Plagiarism seriously. Offenders will be dealt with sternly.

## **Plagiarism Notice**

You are reminded that there exist regulations concerning plagiarism.

### **Extracts from University Regulations on Cheating, Plagiarism and Collusion**

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples .....

- (i) Cheating: including copying coursework.
- (ii) Falsifying data in experimental results.
- (iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
- (iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.
- (v) Collusion to present joint work as the work solely of one individual.
- (vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.
- (vii) Other conduct calculated to secure an advantage on assessment.
- (viii) Assisting in any of the above.

### **Some notes on what this means for students:**

- (i) Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
- (ii) Taking extracts from published sources without attribution is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " $e = mc^2$  (Einstein 1905)". A reference section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Further information in relation to the existing London Metropolitan University regulations concerning plagiarism can be obtained from <http://www.londonmet.ac.uk/academic-regulations>

## Assessment

This assignment will be marked out of 100 and carries 30% of the overall module weighting.

**Your .java files and report for this part must be uploaded and submitted by RTE Deadline.** The assignment must be carried out individually so you must not obtain help from anyone other than the module teaching staff. You must not copy code from any source apart from the module core text and the module materials. Collusion, plagiarism (unreferenced copying), and other forms of cheating constitute Academic Misconduct, which can lead to failure of the module and suspension. The viva will be conducted for this assignment.

**Note: If a student would be unable to defend his/her coursework, s/he might be penalized with 50% of total coursework marks**

## Aim

The aim of this assignment is to implement a real-world problem scenario using the Object-oriented concept of Java that includes creating a class to represent a **Car**, together with its two subclasses to represent a **Fuel Car** and an **Electric Car** respectively. You will also need to write a report that should contain information about your program.

## Deliverables

Create a new project in **BlueJ** and create three new classes (**Car**, **Electric Car**, and **Fuel Car**) within the project. Electric Car and Fuel cars are subclasses of the class Car. When you are ready to submit your solution, upload your codes **Car.java**, **ElectricCar.java**, and **FuelCar.java** files (not any other files from the project) together with your report in pdf format.

## Program (60 marks)

The program should include the following classes (with no additional attributes or methods).

- 1) The **Car** class has five attributes, which correspond to the carID, car name, car brand, car color, and the car price. The car name, car brand, car color, and car price are each represented as a string of text and car id as a number.  
The car id, car name, car brand, and car price are initialized in the constructor by being assigned the value of the constructor's parameters. The car color is initialized with an empty string ("") in the constructor.

Each attribute has a corresponding accessor method.

Define a method to set the car color to a new color by accepting the new color of the car as a parameter.

A display method should output (suitably annotated) the carID, carname, car brand, car color, and car price and, if the car color is empty, the appropriate message should also be displayed.

**[10 marks]**

2) The **FuelCar** class is a subclass of the **Car** class and has eight attributes:

- Distributor Name - a string of characters
- Fuel Type - a string of characters
- Number of Seats - a whole number
- Booked Date - a string of characters
- Purchase Date - a string of characters
- Mileage - a whole number
- Transmission Type- a string of characters
- isPurchased - either true or false (boolean)

The **constructor** accepts six parameters, which are the car id, car name, car brand, car price, fuel type, number of seats, and mileage. A call is made to the superclass constructor with four parameters, the carID, carname, car brand, and car price. Additionally, in the constructor, assign distributor name, booked date, purchase date, and transmission type as an empty ("" ) string, isPurchased status is initialized to false.

Each attribute has a corresponding **accessor method**.

A method is required to **set the distributor name**. The method accepts a **new distributor name** as a parameter. The parameter value is then assigned to the attribute distributor name.

A method is required to **set the transmission type** as different cars have different transmission types. The method accepts a new transmission type as a parameter and assigns the new value to the attribute transmission type.

There is a method to **purchase** a particular fuel car. The method accepts two parameters which are purchase date and booking date. If the car is already

purchased, the method to set the car color is called. Also, an appropriate message including car id, car name, car brand, car color, car price, distributor name should be displayed.

If the car is not purchased yet, the method to set the distributor name and transmission type is called. The parameter values are initialized in the attributes. The status of isPurchased is set to **true**.

A method to **display** the details of the car is required. It must have the same signature as the display method in the car class. It will call the method in the Car class to display the carID, car name, car brand, car color, and the car price. If the fuel car is already purchased then distributor name, fuel type, purchase date, booked date, mileage, number of seats, and transmission type must also be displayed. Each output must be suitably annotated.

**[20 marks]**

- 3) The **ElectricCar** class is also a subclass of the **Car** class and it has eight attributes:

Customer name	- a string
Battery Capacity	- a whole number
Battery Warranty	- a whole number
Purchase date	- a String
Range	- a String
Recharge time	- a whole number
isBought	- either true or false (boolean)
isSold	- either true or false (boolean)

The **constructor** accepts five parameters which are the car id, car name, car brand, car price, and battery capacity. A call is made to the superclass constructor with four parameters, the carID, carname, car brand, and car price. The attribute battery capacity is given the corresponding parameter value. Additionally, the constructor initializes customer name, battery warranty, purchase date, range, and Recharge time to empty ("" ) string, and isBought and isSold status to false.

Each attribute has a corresponding **accessor method**.

A method is required to **set the customer name** as changes to the name of the customer inevitably occur. The method accepts a new customer name as a parameter and, if the electric car is not bought yet, the new value is assigned to

the customer name attribute. If the electric car is already bought, then a suitable message is output to the user indicating that it is therefore not possible to change the customer name.

There is a method to **buy** an electric car. The method has five parameters, the customer name, battery warranty, purchase date, range, and recharge time. If the electric car is not bought yet, the method to set the customer name is called and the status of isbought is set to **true**, otherwise, the suitable message is displayed to the user that indicates the electric car is already bought.

There should also be a method to **sell** the electric car. The method accepts a new customer name as a parameter. Set the customer name with the values as specified in the parameter.

If the status of isSold is **false**, set the **battery capacity, battery warranty, purchase date, range, and recharge time** is set to **""**. Also, set the status of isSold to **true** and isBought to **false**.

If the electric car is already sold, a suitable message should be displayed indicating that the car is already sold.

A method to **display** the details of the electric car is required. It must have the same signature as the display method in the Car class. It will call the method in the Course class to display the Car ID, car name, car brand, car color, and the car price. If the electric car is already bought then the customer's name, battery capacity, battery warranty, purchase date, range, and recharge time must also be displayed. Each output must be suitably annotated. **[20 marks]**

Additional marks will be awarded for good programming styles, particularly naming, layout and comments.

See <http://www.bluej.org/objects-first/styleguide.html> for details.

**[10 marks]**

## **Report (40 marks)**

Your report should describe the process of development of your classes with:

- a. A class diagram **[5 marks]**
- b. Pseudocode for each method in each class **[10 marks]**
- c. A short description of what each method does **[5 marks]**
- d. You should give evidence (through inspection tables and appropriate

screenshots) of the following testing that you carried out on your program:

**Test 1:** Inspect FuelCar class, purchase the FuelCar, and re-inspect the FuelCar Class [2 marks]

**Test 2:** Inspect ElectricCar class, buy an electric car and re-inspect the ElectricCar Class [2 marks]

**Test 3:** Inspect ElectricCar class again, change the status of isBought to true, and re-inspect the ElectricCar class [2 marks]

**Test 4:** Display the details of FuelCar and ElectricCar classes. [4 marks]

- e. The report should contain a section on error detection and error correction where you give examples and evidence of three errors encountered in your implementation. The errors (syntax, semantic or logical errors) should be distinctive and not of the same type. [3 marks]
- f. The report should contain a conclusion, where you need to include the following things:
- Evaluation of your work,
  - Reflection on what you learned from the assignment,
  - What difficulties do you encounter and
  - How you overcame the difficulties.

[4 marks]

The report should include a title page (including your name and ID number), a table of contents (with page numbers), an introduction part that contains a brief about your work, and a listing of the code (in an appendix). Marks will also be awarded for the quality of writing and the presentation of the report.

[3 marks]

**Viva**

**Note:** *If a student would be unable to defend his/her coursework, s/he might be penalized with 50% of total coursework marks.*

## Marking Scheme

Marking criteria		Marks
<b>A.</b>	<b>Coding Part</b>	<b>60 Marks</b>
	<ol style="list-style-type: none"> <li>1. Creating Car Class</li> <li>2. Creating FuelCar Class</li> <li>3. Creating ElectricCar Class</li> <li>4. Program Style</li> </ol>	<div>10 Marks</div> <div>20 Marks</div> <div>20 Marks</div> <div>10 Marks</div>
<b>B.</b>	<b>Report Structure and Format</b>	<b>40 Marks</b>
	<ol style="list-style-type: none"> <li>1. Class Diagram</li> <li>2. Pseudocode</li> <li>3. Method Description</li> <li>4. Test-1</li> <li>5. Test-2</li> <li>6. Test-3</li> <li>7. Test-4</li> <li>8. Error Detection and Correction</li> <li>9. Conclusion</li> <li>10. Overall Report Presentation/Formatting</li> </ol>	<div>5 Marks</div> <div>10 Marks</div> <div>5 Marks</div> <div>2 Marks</div> <div>2 Marks</div> <div>2 Marks</div> <div>4 Marks</div> <div>3 Marks</div> <div>4 Marks</div> <div>3 Marks</div>
<b>Total</b>		<b>100 Marks</b>