Student Name: Muhammad Fahad     Student ID: FA19-BSSE-0014

# Mohammad Ali Jinnah University

## Object Oriented Programming (Lab)

### Final Examination Spring 2020

## Section-AM/BM

**Instructor Name: Samia Bashir** **Total Marks: 50**
**Date: July 1, 2020** **Time: 180 minutes**

**Honor Code Pledge**: "I pledge on my honor that I have not given or received any unauthorized assistance on this examination."

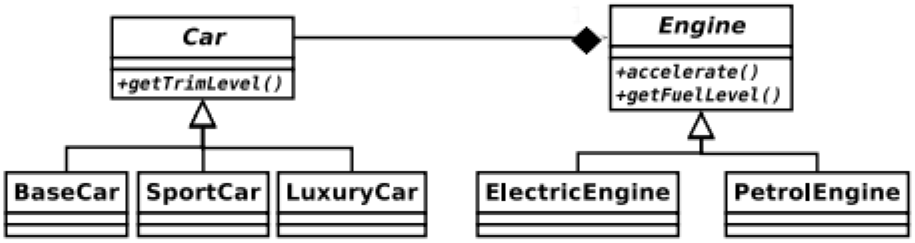
**Instructions**:

1. Write down your student name and ID on the question paper. You have to submit the question paper in pdf format along with your .java files.
2. Draw UML diagrams wherever required on a separate sheet of paper, scan it and turn it in.
3. Write the honor code on top of the paper where you draw your UML and sign it.
4. Plagiarism of any sort will be penalized strictly.
5. Write all constructors, accessors, mutator methods.
6. Attempt all the questions.

**Question # 1**: **Marks: 10**

A car manufacturer uses Java software to track current vehicles being built. The UML diagram below shows an excerpt of the current software structure. You should assume the presence of other appropriate fields and methods. Each car can be built to one of three trim levels: base, luxury or sport. They can also be configured with an electric or petrol engine.

a. Write a program to implement the given scenario.



b. The manufacturer decides to offer a vehicle with a hybrid engine that is both an electric engine and a petrol engine. Make changes in code as well as UML (Draw the new UML) to achieve the effect of multiple inheritance for **HybridEngine**.

c. Write a test class to test your system.

**Question # 2**: **Marks: 10**

The atoms of different elements have different numbers of protons, neutrons and electrons. Electrons are negatively charged, protons are positively charged, and neutrons have no charge.
a) Write an atom class that contains:

(i) fields for storing the numbers of protons, neutrons and electrons with appropriate visibility;
(ii) setter and getter methods for manipulating these fields, ensuring that the minimum value for electrons and protons is 1, and the minimum value for neutrons is 0;
(iii) a constructor that initialises new objects of atom to be the smallest element (Hydrogen), for which the number of protons is 1, the number of neutrons is 0, and the number of electrons is 1.

b) Write a new method for the atom class called isIon() that will return true or false, depending upon whether the atom is an ion. An atom is an ion if it is charged (i.e., if the number of electrons ≠ the number of protons).
c) Write a new method for the atom class called getAtomicMassNumber that will calculate and return the atomic mass number of the atom. Atomic mass number of an atom (often denoted A) is defined as the number of protons plus the number of neutrons.

**Question # 3**: **Marks: 10**

a. Write a java program that creates a file and name it OOPFinal.txt if it does not exist already. In that file, write 20 integers created randomly using appropriate methods/classes.
b. Now read the numbers and display on the screen in ascending order.

**Question # 4**: **Marks: 20**

Sports World is an Events Management Company that organizes the running of major sporting games, such as the Commonwealth Games.
They have hired you to design a system to manage their tasks, you are required to design a few components as a sample first. The entire process is done in the following way:

- Before being used to host a sporting event, venues are assessed by an **`Administrator`** who checks that they are for fit for purpose.
  **`boolean assessVenue(int capacity,boolean safe, boolean healthy)`**

  If the venue can hold more than 10,000 people, the **`Administrator`** conducts additional health and safety checks to ensure that the venue is safe.
- Befor
  e the games begin, a Team of **`Organizers`** are appointed to run the day-to-day operations, including booking successfully assessed/safety-checked venues (**`bookVenue()`**)so only if the venues are approved, the organizers will perform their tasks. The organizers then produce a schedule **produceSchedule**() that shows a list of all the programs. Hint: Use ArrayList

- System also has a **Programme** that lists the **progId**, **progName**, **date(type Date)**, **time** and **location** of each sporting event
- At this point, **Athletes** can register for an event by giving their **name**, **address**, **dateOfBirth of type Date** and their **program**. Program should also **calculateAge()** of Athlete
- Date includes **day**, **month** and **year**.
- Some **OverseasAthletes** need to **applyForVisa()** and the system needs to record whether they were successful.
- At the end of each event, organizer **produceResults()** that records the positions of each athlete. For now, the method should just print a statement.
- Once all the events are completed, organizer declares the **programCompleted()**, the **Administrator** checks the Table of Results for accuracy and **produceMedalTable().** For now, the method should just print a statement.

The test file is given to you. This file should run without errors after you have implemented the above scenario.

**Test file:**

```
import java.util.ArrayList;

public class Test
{
    public static void main(String[] args)
    {
        Date athleteDate= new Date("February",1993,5);
        Date athlete2Date= new Date("November",1990,20);
        Date progDate= new Date("May", 2018,10);
        Programme programme= new Programme(1, "Kabaddi", progDate,"10:00 PM",
"Pakistan");
        Programme programme1= new Programme(2, "Tennis", new
Date("March",2019,2),"12:00","MAJU");
        Athletes[] athletes= new Athletes[2];
        athletes[0]= new Athletes("Sultan ","Lyari", athleteDate, programme);
        athletes[1]= new OverseasAthletes("Sania Mirza ", "Mumbai",athleteDate,
programme1);
        ((OverseasAthletes)athletes[1]).applyForVisa(true);
        for(int i=0;i<athletes.length;i++) {

            System.out.println(athletes[i].toString());
            athletes[i].registerForEvent();
        }
        ArrayList<Programme> programmes= new ArrayList<Programme>();
        programmes.add(programme);
        programmes.add(programme1);

        Admin admin= new Admin();
        if(admin.assessVenue(15000, true, true))
        {
            Organizer organize= new Organizer(programmes);
            organize.bookVenue();
        organize.produceSchedule();
            organize.programCompleted();              organize.produceResultsTable();
            admin.produceMedalTable();

        }
    }

}
```