OOP Assignment-01

- **[1 HOUR] software requirement understanding**
  - reading of the assignment document
  - understanding of problem statement
  - understanding of software inputs and outputs

- **[2 HOURS] software design**
  - identification of user-software interaction sequences
    - software menus
    - user inputs and
    - software flows
  - **identification of required variables and data structures to be used**
    - for input, processing, and output
    - identification of storage class specifications of required variables
  - **modularization of software into required functions**
    - proper naming of functions
    - identification of tasks to be performed into functions
    - identification of parameters and return types of functions
  - **identification of user defined header files**
    - names and functions to be placed in
  - **flow sequence of main function and call of various user defined functions from it**
- **[6 HOURS] software coding**
  - coding of user defined functions
  - placement of user defined functions into header file/s
  - coding of main function
- **[2 HOURS] software testing**
  - running software on various inputs
  - identification of software error and bugs
  - removal of software error and bugs
    - [1 HOUR] software documentation
  - **proper indentation of code**
  - use of proper naming conventions
  - commenting of code

Objectives:

This assignment will provide experience with

- Class
- Constructors
- Member Functions
- Class Composition
- Static arrays
- Dynamic arrays

Prerequisite Skills:

- Understanding of Class design
- Understanding of Constructor
- Understanding of use of Arrays
- Understanding of use of Dynamic Arrays
- Understanding of Class Composition

Problem Statement:

This assignment of yours is mainly based on your creativity. You are given general structure of the requirements but you are not bound. You are free to showcase your creativity and ideas. Your individual ideas, assumptions and efforts will be appreciated.

Basic Structures:

Create a class **Vehicle**. It will have the following:

Properties:

- ID
- Number of Tyre
- Number of Doors
- Color
- Runs on (Diesel or Petroleum) (it describes the vehicles runs on Diesel or Petroleum)

✧ *(You can include any other data members as well if you want to, according to your creativity)*

Methods:

- Make Required Constructors
- Make necessary setters and getters
- Make display function

✧ *(You can include any other member functions as well if you want to, according to your creativity)*

Create another class **Car.** It will have the following properties

- Vehicle object
- Manufacturer name
- Model
- Max speed
- Current speed
- Car type (Family use, Classical or Racing)

(You can include any other data members as well if you want to, according to your creativity)

Methods:

- Make Required Constructors
- Make necessary setters and getters
- Make display function

✧ *(You can include any other member functions as well if you want to, according to your creativity)*

✧ You can create ADTs for more vehicles as well if needed.

Environment:

Understanding:

Environment is a domain or area where the different abstract classes can be used. These abstract classes objects work like agents in the Environment. They play their role and perform actions in the environment. An Environment will have different and multiple abstract classes objects working in it and together they perform a collective Task. An Environment can further be a class where other abstract classes are used in using composition or it can consist of group of functions working together and using abstract classes objects.

Task:

Your Task in this Assignment to implement an Environment of your desire where you will use abstract classes mentioned above. You are not bound to any Environment. Just make sure wherever you use these classes objects, they must be used as agents in that Environment and overall environment must achieve a collective Goal.

You are Free:

For this purpose you will add data members and methods in the above classes according to your requirements in hand (depends upon what kind of Environment you are trying to implement). You are also encouraged to create further classes if needed.

Assessment Criteria:

Your assignment will be assessed according to how well Environment you have implemented and how well you are using the Classes. It will depend upon how you have implemented your creativity and ideas on top of the general required structure. In the end, your whole assignment must showcase an Environment that performs a unique Task.

Submission:

Please place different Classes in different files. For example if you have Car class if will come in Car.h header file. Then you will include Car.h in main.cpp file. So like this, you will make multiple files, in your assignment folder.

Submit your folder in Zip form.

Cheating and AI use Policy:

For this assignment, Google Classroom's Plagiarism and AI detection service will be used. So you are prohibited to any kind of Copying or AI use.