**Description:**

The B&M company sells cars. They want to create a software system that manages their employees and the cars that they sell. A part of their software requirement is given below. You are required to do self-study on how the car sales function and create a design for managing the functions of the company. In addition to the requirements given below, you are encouraged to add more attributes and functionalities to the system.

**Requirements:**

The company has different employees who are either Mangers or Salespersons.  An example of a partial file of employees is given below. The personal details of the employees like their age, date of birth, passport details are also stored in the system. Each manager has a group of salespersons assigned to work with her/him. For each sale that a salesperson makes, 6.5% of the profit goes to the salesperson, 3.5% of the profit goes to the manager and the rest to the company. The profit is the amount that the salesperson makes over and above the price of the car.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **ID Number** | **Department** | **Job Title** | **Basic Salary** |
| Susan Meyers | 47899 | Accounting | Manager | 37500 |
| Mark Jones | 39119 | IT | Salesperson | 26000 |
| Joy Rogers | 81774 | Manufacturing | Salesperson | 24000 |

The company has three types of cars, hatchback, sedan, and SUV. An example of a partial file of cars is given below. Other details of each car, like the fuel capacity, max speed, and color are also stored in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID Number** | **Price** | **Type** |
| Jazz | VX3 | 55000 | Hatch |
| Mark3 | SX3 | 84000 | Sedan |
| Wagoner | ZX3 | 125000 | SUV |

Your program must have the following functions:

1. The system should be able to add/delete/modify details of employees and cars.
2. Display all the details of the employees, given the ID number.
3. Display all the details of a car, given the ID number.
4. Display sales details of any employee, given the ID number
5. All details must be stored in files and any wrong or exceptional input must be appropriately handled.
6. From the above example, assume that Joy Rogers and Mark Jones are working with Susan Meyers. Mark sells 2 hatchback cars and 3 SUVs this month. Joy sells 1 SUV, 2 hatchbacks and 2 sedans this month. Your program should show the salaries that all three of them would get this month after they make the sale.

|  |  |  |
| --- | --- | --- |
| **Name** | **Car Sold** | **Sale Price** |
| Joy Rogers | ZX3 | 155000 |
| Joy Rogers | VX3 | 57800 |
| Joy Rogers | VX3 | 55000 |
| Joy Rogers | SX3 | 89000 |
| Joy Rogers | SX3 | 93000 |
| Mark Jones | VX3 | 58000 |
| Mark Jones | VX3 | 58000 |
| Mark Jones | VX3 | 158000 |
| Mark Jones | VX3 | 158000 |
| Mark Jones | VX3 | 158000 |

**Submission:**

* Submit a report that has the following:
  + A paraphrased description of the system based on the requirements given above and a self-study of the required system.
  + A list of all classes (related attributes and behaviors), relationships between classes, and assumptions made.
  + UML class diagram with all class relationships included.
  + Python code that represents classes, which includes the constructor, setter/getter, and other functions for the given requirements.
  + 5% of the total score will be allocated for good documentation of the code and timely submission

**Report format to submit**

* Title Page: Include case-study title, student ID, and full name
* Problem Analysis (20%): In this section, based on given requirements and self-study, a detailed list of all the requirements is given. The list of all classes, their attributes, and behaviors are also listed with data types.
* Functional Design (20%): In this section, the algorithm or flow chart is provided to explain the logical flow that will drive the use of the system. File structure and information stored in files are described.
* Class Design (30%): The UML class diagram, with class relationships, and cardinality for the business case is provided. Each relationship is explained, and assumptions are listed.
* Python Code (20%): In this section, provide the Python class structures for all the identified classes with required functionalities. The test code of the system is NOT required to be submitted.
* Conclusion (5%): In this section include a reflection on what was learned in this exercise, the challenges faced while working on this assignment, and how the system can be further expanded.